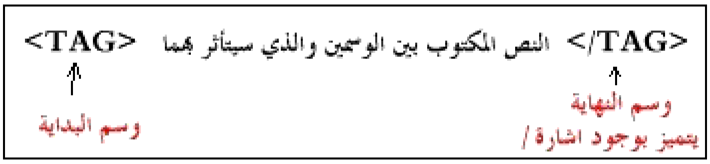
المقدمة :

المقدمة

إنها اللغة المستخدمة لإنشاء صفحات الإنترنت. (والكلمة إختصار لــHyper Text Markup Language).
وهي ليست لغة برمجة بالمعنى والشكل المتعارف عليه للغات البرمجة الأخرى كلغة C . فهي مثلاً لا تحتوي على جمل التحكم والدوران، وعند الحاجة لاستخدام هذه الجمل يجب تضمين شيفرات من لغات أخرى كـــ Java, JavaScript, CGI . كذلك فهي لا تحتاج إلى مترجم خاص به Compiler . وهي غير مرتبطة بنظام تشغيل معين، لأنه يتم تفسيرها وتنفيذ تعليماتها مباشرة من قبل متصفح الإنترنت وبغض النظر عن النظام المستخدم. لذلك فهي لغة بسيطة جداً، وسهلة الفهم والتعلم ولا تحتاج لمعرفة مسبقة بلغات البرمجة والهيكلية المستخدمة فيها. بل ربما كل ما تحتاجه هو القليل من التفكير المنطقي وترتيب الأفكار.

تتكون مفردات لغة Html من شيفرات تسمى TAGS أي الوسوم. وهي تستخدم بشكل أزواج وتكتب بالصيغة التالية (من اليسار إلى اليمين) :-



فعلى سبيل المثال الوسم <B> يستخدم لكتابة الكلمات بخط أسود عريض Bold وذلك بالشكل التالي:

</B> Text <B>

وهناك بعض الوسوم الخاصة التي تستخدم بصورة مفردة مثل وسم نهاية السطر
 أو قد تستخدم بكلتا الحالتين مثل وسم الفقرة <P>.
وسوف نناقش هذه الوسوم وغيرها بالتفصيل في حينه إن شاء الله

كيف نبدأ...

لا يتطلب كتابة ملف HTML أية برامج خاصة فهي كما قلنا لغة لا تحتوي على برنامج مترجم. بل نحتاج فقط إلى برنامج لتحرير النصوص البسيطة ومعالجتها، وبرنامج المفكرة الموجود في Windows يفي بهذا الغرض. وكذلك إلى أحد متصفحات الإنترنت Netscape Navigator أو MS Internet Explorer لمعاينة الصفحات التي نقوم بتصميمها. وعليك فقط أن نقوم بحفظ النص المكتوب بملف يحمل الاسم الممتد html. أو htm.
والجدير ذكره أنه يوجد العديد من البرامج التي تستخدم لإنشاء صفحات Html. دون الحاجة لمعرفة هذه اللغة حيث يقوم المستخدم من خلالها بكتابة الصفحات وتصميمها بما تحويه من نصوص ورسومات وجداول ثم يقوم البرنامج بتخليق الوسوم المناسبة وتحويل هذه الصفحات من وراء الكواليس تلقائياً وحفظها بتنسيق html. أي أن دور المستخدم ينحصر في الكتابة والتصميم فقط، دون معرفته للشيفرة التي استخدمت. وبالتالي عدم قدرته على التحكم بأي وسم أو تعديل الشيفرة حسب الحاجة، إلا من خلال إعادته للتصميم الأساسي ثم إعادة التحويل والحفظ من قبل البرنامج. وهذه الطريقة على سهولتها وسرعتها نسبياً، إلا أني لا أنصح باستخدامها لمن يريد معرفة هذه اللغة والتمكن منها.

قبل أن نبدأ : حسناً، لديك محرر نصوص ممتاز لكتابة ملفات HTML ولديك متصفح إنترنت لمعاينتها، ولديك هذه الدروس التي ستنطلق معها إلى عالم تصميم صفحات الويب. هل هذا يكفي؟ برأيي المتواضع، لا.

تحتاج دائماً وأبداً إلى تطبيق ما تتعلمه بصورة عملية أكثر من مجرد الأمثلة المدرجة في الدروس. ما رأيك في أن تفكر بموضوع ما يستهويك وتحب أن تتعاطى به؟ وتخيل أنك ستقوم بإنشاء موقع ويب عنه بصورة واقعية. ومع تقدمك في الدروس قم بتطبيق ما فيها على صفحاتك. ستجد الكثير من المتعة في هذا، وستسر جداً عندما ترى صفحتك تتبلور أمام عينيك يوماً بعد يوم، والأهم من هذا كله هو أنك ستكتشف أي ثغرات في استيعابك لهذه الدروس (وعندها من المؤكد أنك ستقوم بتلافيها) وقد تكتشف كذلك ثغرات ارتكبها كاتب هذه الدروس.

والآن... لنبدأ

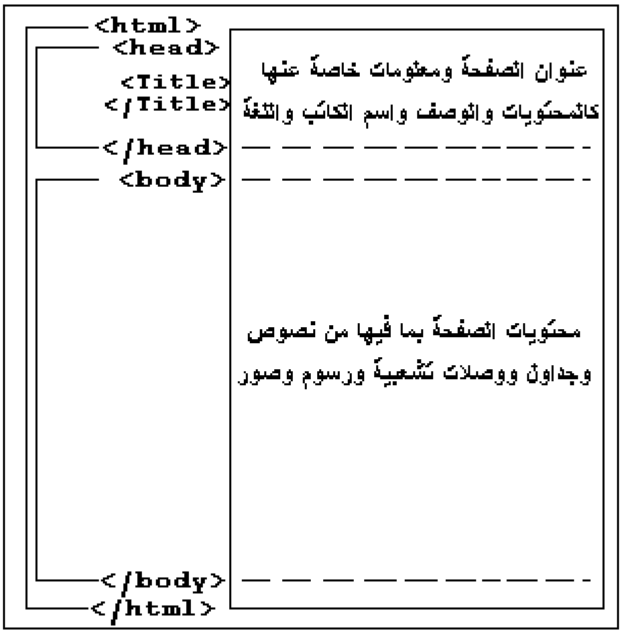
الدرس الأول
الأساسيات

أهلاً وسهلاً بك إلى الدرس الأول من دروس HTML. سوف أقوم في هذا الدرس بسرد الوسوم الأساسية لصفحة الويب ومناقشتها معك واحداً تلو الآخر. لنصل في النهاية إلى إنشاء صفحة ويب بسيطة.

لنأخذ الوسوم التالية:

وسم البداية	وسم النهاية
<HTML>	</HTML>
<HEAD>	</HEAD>
<TITLE>	</TITLE>
<BODY>	</BODY>

ماذا تلاحظ؟ أن كل منها يتألف من زوج من الوسوم أحدهما وسم البداية، والآخر وسم النهاية. ويتميز وسم النهاية بوجود الرمز / . تأمل الرسم التالي، فهو يعطي فكرة عن تركيب ملف Html



☜ إذن فملف Html يبدأ دائماً بالوسم <HTML> وينتهي بالوسم </HTML>. لا تنسى ذلك!

أما الوسم <HEAD> فيحدد بداية المقطع الذي يحتوي على المعلومات الخاصة بتعريف الصفحة. كالعنوان الظاهر على شريط عنوان المتصفح. وهذا العنوان بدوره يحتاج لأن يوضع بين الوسمين: <TITLE> ... </TITLE> وبالطبع يجب كتابة الوسم </HEAD> لكي ننهي هذا المقطع.

نأتي إلى الوسم <BODY> والذي يتم كتابة نصوص صفحة الويب ضمنه، بالإضافة إلى إدراج الصور والجداول وباقي محتويات الصفحة. وهو أيضاً يحتاج إلى وسم الإنهاء </BODY>

ما رأيك لو نبدأ بتطبيق هذه المعلومات بصورة عمليه؟ هيا... قم بفتح برنامج المفكرة واكتب ما يلي:

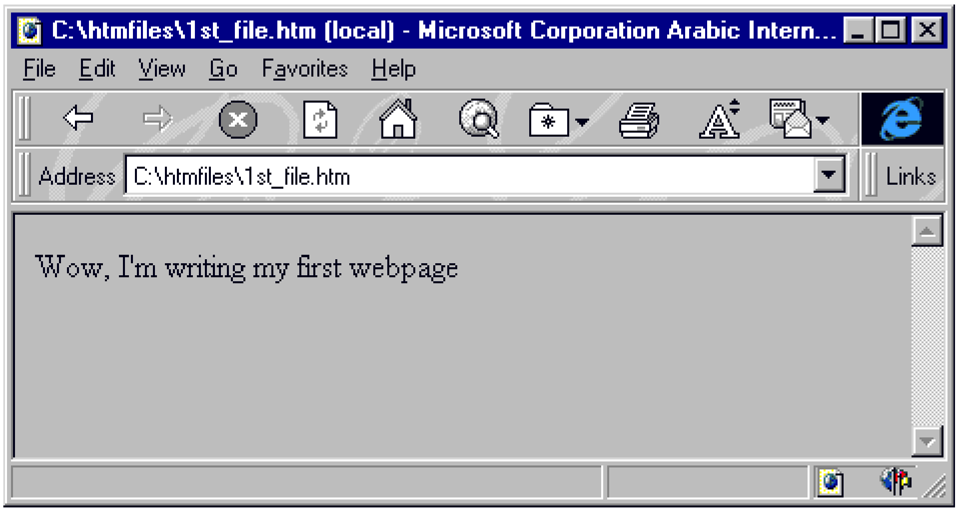
```
<HTML>
<HEAD>
<TITLE>
This is a test Webpage
</TITLE>
</HEAD>

<BODY>
Wow, I'm writing my first webpage
</BODY>
</HTML>
```

والآن قم بحفظ ما كتبته في ملف وبأي اسم تختاره. ولا تنسى أن الامتداد المستخدم في أسماء ملفات HTML هو *.htm* أو *.html* مثلاً أنا اخترت الاسم 1st_file.htm ومن الأفضل أن تقوم بإنشاء مجلد مستقل على القرص الصلب لكي تحفظ به ملفاتك فهذا يسهل عليك عملية استرجاعها للعرض أو التحديث.

حان وقت العرض، لكي نشاهد نتيجة ما كتبناه. قم بتشغيل متصفح الإنترنت الذي تستخدمه. فإذا كان Netscape Navigator اختار الأمر File Open... من قائمة File. أما في MS Internet Explorer فاختار الأمر Open... من قائمة File. ثم حدد المسار الذي يوجد به الملف.

وذلك طبعاً حسب الافتراضات السابقة التي اتبعتها عند تخزين الملف. وهذا ما حصلت عليه:



هل حصلت على نفس النتيجة؟ إذن مبروك ☺ لقد قمت بإنشاء أول صفحة ويب خاصة بك.

وقبل أن نستمر أريد أن أنبهك إلى بعض الملاحظات عند كتابة صفحات الويب:

- لا يوجد فرق بين كتابة الوسوم بالأحرف الإنجليزية الكبيرة UPPERCASE أو الأحرف الصغيرة lowercase. لذلك تستطيع الكتابة بأي شكل منهما أو حتى الكتابة بكليهما.

- إن المتصفحات لا تأخذ بعين الاعتبار الفراغات الزائدة أو إشارات نهاية الفقرات (أي عندما تقوم بضغط مفتاح *Enter*) التي تجدها هذه المتصفحات في ملف Html. وبعبارة أخرى فإن باستطاعتك كتابة ملفك السابق بالشكل التالي:

```
<HTML><HEAD><TITLE> This is a test Webpage </TITLE></HEAD><BODY>
Wow, I'm writing my first webpage </BODY></HTML>
```

أو بالشكل التالي:

```
<HTML>
<HEAD>
<TITLE>
This
is a
test
Webpage
</TITLE>
</HEAD>
<BODY>
Wow,
I'm
writing
my
first
webpage
</BODY>
</HTML>
```

أو حتى بهذا الشكل:

```
<HTML> <HEAD> <TITLE>
This is a test Webpage
</TITLE>
</HEAD>
<BODY>
Wow, I'm writing my first webpage
</BODY>
</HTML>
```

وفي كل الحالات ستحصل على نفس النتيجة.
لكن هذا لا يعني أن الفقرة المكونة مثلاً من عشرة أسطر ستمتد إلى عدة أمتار بعرض الشاشة. كلا بالطبع لأن المتصفح سيقوم بعمل التفاف تلقائي لها بحسب عرض الشاشة، مهما كان مقدار هذا العرض.
والآن قد تتساءل، إذن كيف يمكن التحكم بمقدار النص المكتوب في كل سطر وكيف يمكن تحديد نهاية الفقرة وبداية الفقرة التي تليها؟ سؤال وجيه!!! والإجابة عليه هي:

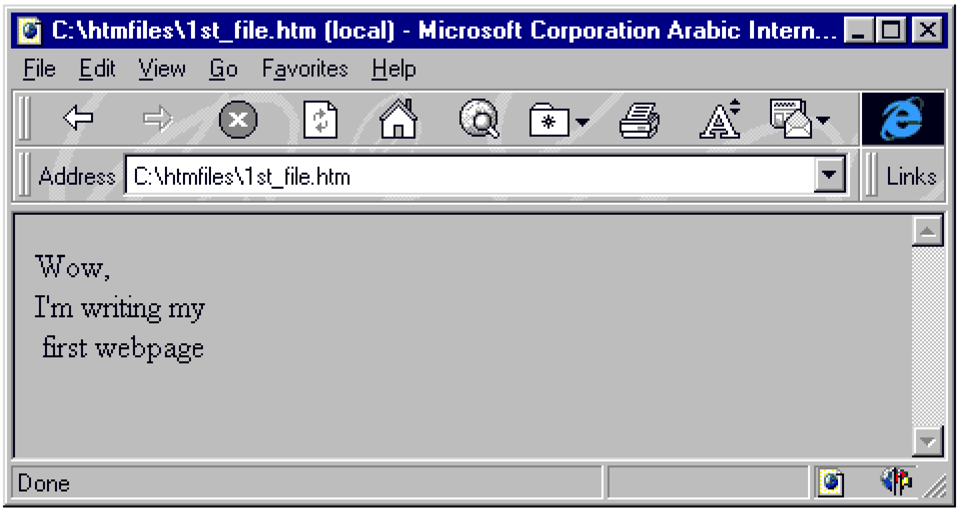
سوف نستخدم الوسم
 لتحديد النهاية للسطر. والبدء بسطر جديد (لاحظ أن هذا الوسم مفرد، أي ليس له وسم نهاية).

ونعود إلى المثال السابق، قم بتعديل الملف لكي يصبح بالشكل التالي

```
<HTML>
<HEAD>
<TITLE>
This is a test Webpage
</TITLE>
```

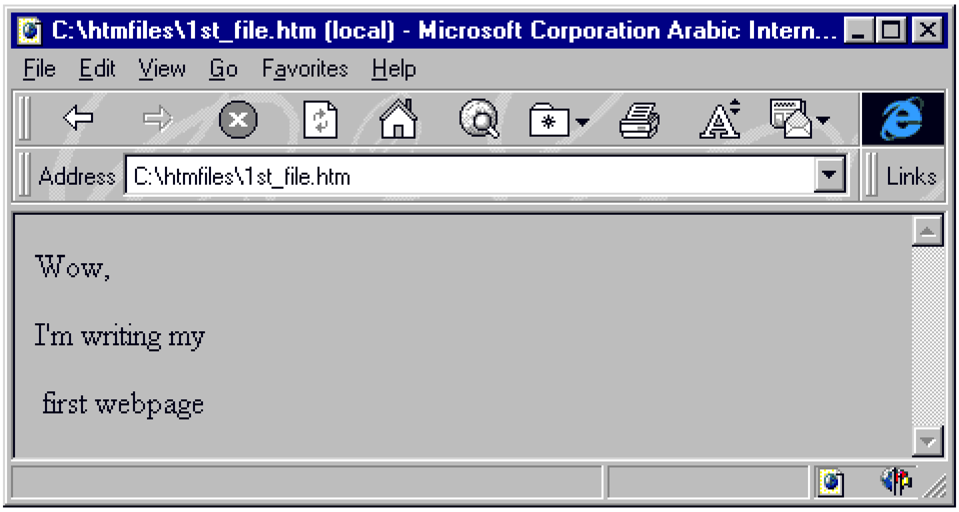
```
</HEAD>
<BODY>

Wow, <BR> I'm writing my <BR>
first webpage
</BODY>
</HTML>
```



وهناك أيضا الوسم <P> الذي يقوم تقريباً بنفس عمل الوسم السابق أي أنه ينهي السطر أو الفقرة ويبدأ بسطر جديد لكن مع إضافة سطر إضافي فارغ بين الفقرات.

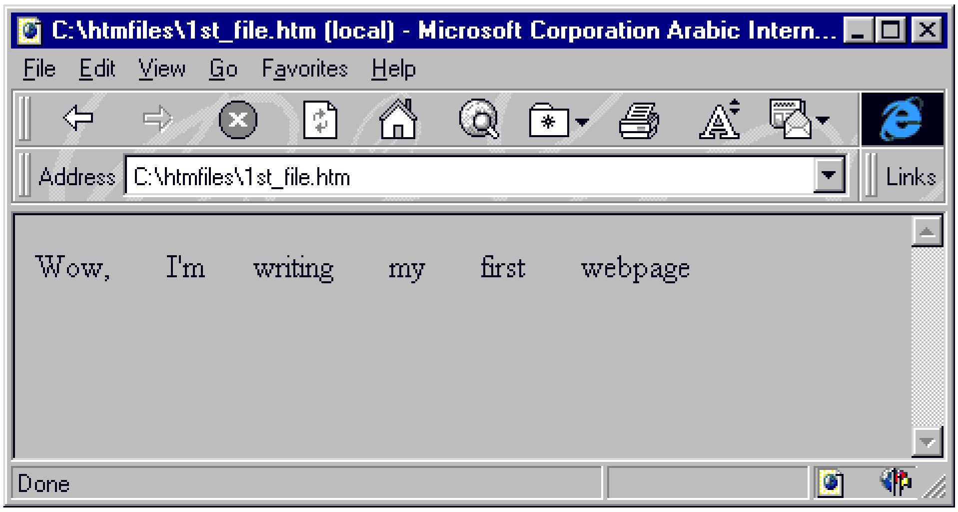
```
<HTML>
<HEAD>
<TITLE>
This is a test Webpage
</TITLE>
</HEAD>
<BODY>
Wow, <P> I'm writing my <P>
first webpage
</BODY>
</HTML>
```



أما الفراغات فتعتبر رموزاً خاصة لذلك لا نستطيع التحكم بها وبعددها إلا باستخدام الوسم   (والأحرف هي اختصار للعبارة Non Breakable Space). وإذا أردت إدخال عدة فراغات بين نص وآخر ما عليك إلا كتابة هذا الوسم بنفس عدد الفراغات المطلوب. كما يجب عليك التقيد بالأحرف الصغيرة هنا.

إذن لنعد إلى المفكرة ونكتب ملفا بالشكل التالي:

```
<HTML>
<HEAD>
<TITLE>
This is a test Webpage
</TITLE>
</HEAD>
<BODY>
Wow,      
I'm      writing    
  my       first       webpage
</BODY>
</HTML>
```



وبمناسبة الحديث عن الرموز الخاصة فهناك العديد من هذه الرموز والتي يجب أن تكتب بصورة معينة وباستخدام الوسوم وليس مباشرة بصورتها العادية. خذ مثلا إشارتي *أكبر من وأصغر من وإشارة الاقتباس "* . كل هذه الإشارات تستخدم أصلاً مع الوسوم فهي محجوزة ضمن مفردات لغة HTML ومن الخطأ استخدامها بصورتها الصريحة لئلا يؤدي ذلك إلى حدوث مشاكل في طريقة عرض الصفحة. كذلك فإن هناك رموزاً غير موجودة أساساً على لوحة المفاتيح كرمز حقوق الطبع © ورمز العلامة المسجلة ® ونحتاج إلى هذه الطريقة(طريقة الوسوم) لكتابتها. وإليك جدول ببعض هذه الرموز ووسومها المكافئة. وألفت نظرك إلى أنها تكتب كما هي في الجدول وبدون إشارتي <>

الرمز	الكود
	الـــرمـــوز
‘	‘
’	’
‚	‚
“	“
”	”

„	„
†	†
‡	‡
‰	‰
‹	‹
›	›
♠	♠
♣	♣
♥	♥
♦	♦
‾	‾
←	←
↑	↑
→	→
↓	↓
™	™
&	&
<	<
>	>
"	"
×	×
÷	÷
–	–
—	—
¡	¡
¢	¢
£	£
¤	¤
¥	¥
€	€
¦	¦
§	§
▯	&#127;
¨	¨
©	©
ª	ª
«	«
¬	¬
®	®
¯	¯
°	°
±	±
²	²
³	³

´	´
µ	µ
¶	¶
·	·
¸	¸
¹	¹
º	º
»	»
¼	¼
½	½
¾	¾
¿	¿
الحروف Capital	
À	À
Á	Á
Â	Â
Ã	Ã
Ä	Ä
Å	Å
Æ	Æ
Ç	Ç
È	È
É	É
Ê	Ê
Ë	Ë
Ì	Ì
Í	Í
Î	Î
Ï	Ï
Ð	&ET;
Ñ	Ñ
Ò	Ò
Ó	Ó
Ô	Ô
Õ	Õ
Ö	Ö
Ø	Ø
Ù	Ù
Ú	Ú
Û	Û
Ü	Ü
Ý	Ý
Þ	Þ
	الحروف Small
ß	ß

à	à
á	á
â	â
ã	ã
ä	ä
å	å
æ	æ
ç	ç
è	è
é	é
ê	ê
ë	ë
ì	ì
í	í
î	î
ï	ï
ð	&et;
ñ	ñ
ò	ò
ó	ó
ô	ô
õ	õ
ö	ö
ø	&oslas;
ù	ù
ú	ú
û	û
ü	ü
ý	ý
þ	&torn;
ÿ	ÿ

الدرس الثاني
الألوان – أضف لصفحتك بعض الحيوية

أهلاً وسهلاً بك إلى الدرس الثاني من دروس HTML. سوف نقوم في هذا الدرس بالتعرف على الخصائص التي يمكن إضافتها إلى الوسم <BODY> من أجل التحكم بالشكل العام للصفحة، وخصوصا فيما يتعلق بالألوان.

طبعاً أنت لا زلت تذكر الصفحة التي قمنا بكتابتها في الدرس الأول. صفحة بسيطة بخلفية رمادية وخط صغير نسبياً لونه أسود. وهذه هي الإعدادات الإفتراضية التي يعتمدها المتصفح عندما لا نقوم نحن بتحديد إعدادات أخرى. (ربما تقول: أهذه صفحة إنترنت! أين الألوان والرسومات والخطوط الجميلة والتنسيقات التي نراها في صفحات الإنترنت؟ معك حق لكن مهلاً فما زلنا في البداية).
سوف نستمر باستخدام صفحتنا هذه لتوضيح أمثلة هذا الدرس أيضاً، لكن لن أقوم بتكرار كتابة وسوم البداية طالما أن عملنا يتركز في الجزء المخصص لمحتويات الصفحة نفسها أي ضمن الوسمين <BODY> ... </BODY>.
إذن لنبدأ العمل!

نطلق كلمة خاصية (Attribute) على التعابير التي تضاف إلى الوسوم، من أجل تحديد الكيفية أو الشكل الذي تعمل بها هذه الوسوم. وبعبارة أخرى فإن الوسم يقوم بإخبار المتصفح عن العمل الذي يجب القيام به أما الخاصية فتحدد الكيفية التي سيتم بها أداء هذا العمل.

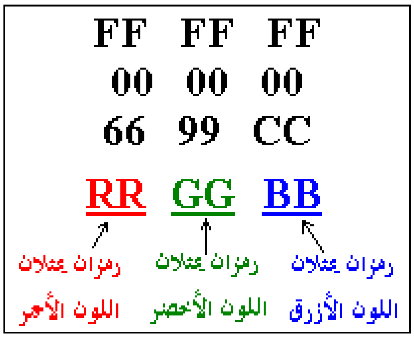
تأمل الشيفرة التالية:

```
<BODY BGCOLOR="FFFFFF">
...
</BODY>
```

لقد قمت بإضافة الخاصية BGCOLOR إلى الوسم <BODY> ، وهي تقوم بتحديد لون الخلفية للصفحة. أما FFFFFF فهي القيمة التي تمثل اللون المختار وهو هنا اللون الأبيض، (لاحظ أنها مكتوبه بين إشارتي " ") ولو أردت تمثيل اللون الأسود لكتبت الرمز 000000. أو الرمز CC6699 للون الأزرق الفاتح......
فمن أين جاءت هذه القيم، وكيف؟... تابع القراءة وسوف تعرف

القليل عن الألوان...

تلاحظ أن القيم السابقة مكونة من ستة رموز، وهي مكتوبة بالصيغة التالية:-



هناك ثلاثة ألوان أساسية هي الأحمر والأخضر والأزرق، ولكل منها يوجد 256 درجة لونية ويعبر عن هذه الدرجات بالأرقام من 000 وحتى 255. ومن خلال مزج هذه الألوان بدرجاتها اللونية المختلفة نحصل على الألوان الأخرى.

* إن أي لون هو مزيج –وبنسبة معينة من الدرجات– من هذه الألوان الثلاثة *

فمثلا اللون الأسود مكون من الدرجة 000 من كل من اللون الأحمر والأخضر والأزرق. واللون الأبيض مكون من الدرجة 255 من هذه الألوان. أما اللون الأصفر فهو مكون من الدرجة 255 للون الأحمر، والدرجة 255 للون الأخضر، والدرجة 000 من اللون الأزرق... وهكذا بنفس الطريقة يتم تكوين باقي الألوان.

وبعملية حسابية بسيطة 256×256×256 ينتج لدينا أن عدد الألوان التي يمكن الحصول عليها بمزج الألوان الثلاثة السابقة هو 16777216 بالضبط.

حسناء، لكن من أي جاءت الرموز FFFFFF والتي عبرت عن اللون الأبيض بها. إنها ببساطة أرقام... مكتوبة بالنظام السداس عشري (نظام عددي أساسه الرقم 16 ويعبر عنه باستخدام الأرقام العادية من 0 إلى 9 والرموز A,B,C,D,E,F). فالرقم 255 بالنظام العشري العادي يكافئه الرقم FF بالنظام السداس عشري.
إذن فالرقم السداس عشري FF على اليسار يمثل الدرجة 255 للون الأحمر. والرقم FF في الوسط يمثل الدرجة 255 من اللون الأخضر. والرقم FF على اليمين يمثل الدرجة 255 من اللون الأزرق.
وعلى هذا المنوال يعبر عن اللون الأزرق الفاتح بالرقم السداس عشري: CC6699 أما اللون الأسود فرقمه هو 000000.

وهذا جدول ببعض الألوان ورموزها المكافئة بالنظام السداس عشري.

Aqua **#00FFFF**	**Antiquewhite** **#FAEBD7**	**Aliceblue** **#F0F8FF**
Beige **#F5F5DC**	**Azure** **#F0FFFF**	**Aquamarine** **#7FFFD4**
Blanchedalmond **#FFEBCD**	**Black** **#000000**	**Bisque** **#FFE4C4**
Brown **#A52A2A**	**Blueviolet** **#8A2BE2**	**Blue** **#0000FF**
Chartreuse **#7FFF00**	**Cadetblue** **#5F9EA0**	**Burlywood** **#DEB887**
Cornflowerblue **#6495ED**	**Coral** **#FF7F50**	**Chocolate** **#D2691E**
Cyan **#00FFFF**	**Crimson** **#DC143C**	**Cornsilk** **#FFF8DC**

Darkgoldenrod #B8860B	Darkcyan #008B8B	Darkblue #00008B
Darkkhaki #BDB76B	Darkgreen #006400	Darkgray #A9A9A9
Darkorange #FF8C00	Darkolivegreen #556B2F	Darkmagenta #8B008B
Darksalmon #E9967A	Darkred #8B0000	Darkorchid #9932CC
Darkslategray #2F4F4F	Darkslateblue #483D8B	Darkseagreen #8FBC8F
Deeppink #FF1493	Darkviolet #9400D3	Darkturquoise #00CED1
Dodgerblue #1E90FF	Dimgray #696969	Deepskyblue #00BFFF
Forestgreen #228B22	Floralwhite #FFFAF0	Firebrick #B22222
Ghostwhite #F8F8FF	Gainsboro #DCDCDC	Fuchsia #FF00FF
Gray #808080	Goldenrod #DAA520	Gold #FFD700
Honeydew #F0FFF0	Greenyellow #ADFF2F	Green #008000
Indigo #4B0082	Indianred #CD5C5C	Hotpink #FF69B4
Lavender #E6E6FA	Khaki #F0E68C	Ivory #FFFFF0
Lemonchiffon #FFFACD	Lawngreen #7CFC00	Lavenderblush #FFF0F5
Lightcyan #E0FFFF	Lightcoral #F08080	Lightblue #ADD8E6
Lightgrey #D3D3D3	Lightgreen #90EE90	Lightgoldenrodyellow #FAFAD2
Lightseagreen #20B2AA	Lightsalmon #FFA07A	Lightpink #FFB6C1
Lightsteelblue #B0C4DE	Lightslategray #778899	Lightskyblue #87CEFA
Limegreen #32CD32	Lime #00FF00	Lightyellow #FFFFE0
Maroon #800000	Magenta #FF00FF	Linen #FAF0E6
Mediumorchid #BA55D3	Mediumblue #0000CD	Mediumauqamarine #66CDAA
Mediumslateblue #7B68EE	Mediumseagreen #3CB371	Mediumpurple #9370D8
Mediumvioletred #C71585	Mediumturquoise #48D1CC	Mediumspringgreen #00FA9A
Mistyrose #FFE4E1	Mintcream #F5FFFA	Midnightblue #191970
Navy #000080	Navajowhite #FFDEAD	Moccasin #FFE4B5

Olivedrab #688E23	Olive #808000	Oldlace #FDF5E6
Orchid #DA70D6	Orangered #FF4500	Orange #FFA500
Paleturquoise #AFEEEE	Palegreen #98FB98	Palegoldenrod #EEE8AA
Peachpuff #FFDAB9	Papayawhip #FFEFD5	Palevioletred #D87093
Plum #DDA0DD	Pink #FFC0CB	Peru #CD853F
Red #FF0000	Purple #800080	Powderblue #B0E0E6
Saddlebrown #8B4513	Royalblue #4169E1	Rosybrown #BC8F8F
Seagreen #2E8B57	Sandybrown #F4A460	Salmon #FA8072
Silver #C0C0C0	Sienna #A0522D	Seashell #FFF5EE
Slategray #708090	Slateblue #6A5ACD	Skyblue #87CEEB
Steelblue #4682B4	Springgreen #00FF7F	Snow #FFFAFA
Thistle #D8BFD8	Teal #008080	Tan #D2B48C
Violet #EE82EE	Turquoise #40E0D0	Tomato #FF6347
Whitesmoke #F5F5F5	White #FFFFFF	Wheat #F5DEB3
	YellowGreen #9ACD32	Yellow #FFFF00

بيان بألوان الخلفيات

#00FF00	#00CC00	#009900	#006600	#003300	#000000
#00FF33	#00CC33	#009933	#006633	#003333	#000033
#00FF66	#00CC66	#009966	#006666	#003366	#000066
#00FF99	#00CC99	#009999	#006699	#003399	#000099
#00FFCC	#00CCCC	#0099CC	#0066CC	#0033CC	#0000CC
#00FFFF	#00CCFF	#0099FF	#0066FF	#0033FF	#0000FF
#33FF00	#33CC00	#339900	#336600	#333300	#330000
#33FF33	#33CC33	#339933	#336633	#333333	#330033
#33FF66	#33CC66	#339966	#336666	#333366	#330066
#33FF99	#33CC99	#339999	#336699	#333399	#330099
#33FFCC	#33CCCC	#3399CC	#3366CC	#3333CC	#3300CC
#33FFFF	#33CCFF	#3399FF	#3366FF	#3333FF	#3300FF

#66FF00	#66CC00	#669900	#666600	#663300	#660000
#66FFCC	#66CC33	#669933	#666633	#663333	#660033
#66FF66	#66CC66	#669966	#666666	#663366	#660066
#66FF99	#66CC99	#669999	#666699	#663399	#660099
#66FFCC	#66CCCC	#6699CC	#6666CC	#6633CC	#6600CC
#66FFFF	#66CCFF	#6699FF	#6666FF	#6633FF	#6600FF
#99FF00	#99CC00	#999900	#996699	#993300	#990000
#99FF33	#99CC33	#999933	#996633	#993333	#990033
#99FF66	#99CC66	#999966	#996666	#993366	#990066
#99FF99	#99CC99	#999999	#996699	#993399	#990099
#99FFCC	#99CCCC	#9999CC	#9966CC	#9933CC	#9900CC
#99FFFF	#99CCFF	#9999FF	#9966FF	#9933FF	#9900FF
#CCFF00	#CCCC00	#CC9900	#CC6600	#CC3300	#CC0000
#CCFF33	#CCCC33	#CC9933	#CC6633	#CC3333	#CC0033
#CCFF66	#CCCC66	#CC9966	#CC6666	#CC3366	#CC0066
#CCFF99	#CCCC99	#CC9999	#CC6699	#CC3399	#CC0099
#CCFFCC	#CCCCCC	#CC99CC	#CC66CC	#CC33CC	#CC00CC
#CCFFFF	#CCCCFF	#CC99FF	#CC66FF	#CC33FF	#CC00FF
#FFFF00	#FFCC00	#FF9900	#FF6600	#FF3300	#FF0000
#FFFF33	#FFCC33	#FF9933	#FF6633	#FF3333	#FF0033
#FFFF66	#FFCC66	#FF9966	#FF6666	#FF3366	#FF0066
#FFFF99	#FFCC99	#FF9999	#FF6699	#FF3399	#FF0099
#FFFFCC	#FFCCCC	#FF99CC	#FF66CC	#FF33CC	#FF00CC
#FFFFFF	#FFCCFF	#FF99FF	#FF66FF	#FF33FF	#FF00FF

أما كيف تعرف الرمز الخاص باللون الذي تريد اختياره، فيوجد برامج خاصة تستطيع من خلالها دمج الألوان الثلاثة بنسب مختلفة، ومن ثم يقوم البرنامج بتوليد الرمز السداس عشري المكافئ للون الناتج.

ملاحظة مهمة:
بعض المتصفحات لا تتعرف على رموز الألوان إلا بوضع إشارة # قبل هذه الرموز، لذلك من الأفضل استخدامها دائماً.

وبالنسبة لبعض الألوان الأساسية والدارجة، من الممكن استخدام أسماء هذه الألوان مباشرة بدلاً من الأرقام السداس عشرية. وهذا جدول يوضح هذه الألوان ومسمياتها:

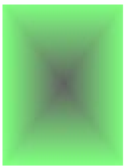
White		Black	
Green		Red	
Purple		Marron	
Blue		Navy	

Lime		Teal	
Silver		Gray	
Aqua		Olive	
Yellow		Fuchsia	

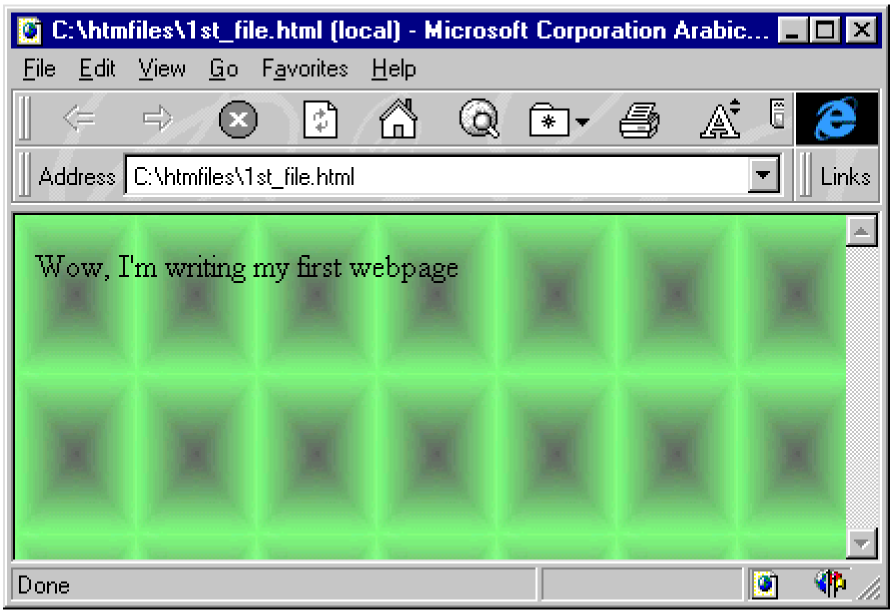
ونعود إلى الوسوم و خصائصها ...

```
<BODY BGCOLOR="#FFFFFF" BACKGROUND="image.jpg">
...
</BODY>
```

تقوم الخاصية BACKGROUND بتحديد صورة كخلفية (ورق جدران) للصفحة وقد استخدمت الصورة التالية:



والمسماة image.jpg في صفحتي وكانت هذه النتيجة



تلاحظ أن المتصفح قد قام بتكرار عرض الصورة بطريقة التجانب وأنها أصبحت تغطي كل الشاشة. بحيث حجبت أيضاً اللون الأبيض الذي حددناه كلون الخلفية (من خلال الخاصية BGCOLOR) والحقيقة أن اللون يظهر فقط عندما

لا نقوم باستخدام صورة ما كخلفية. ومع ذلك يفضل تحديده إحتياطاً خاصة وأن بعض المتصفحات القديمة توصف بأنها متصفحات نصية Text-Based Browsers (أي ليس بإمكانها عرض الصور). أو ربما هناك بعض المستخدمين الذين قاموا بإلغاء خيار عرض الصور تلقائياً من متصفحاتهم. إذن لنعطهم على الأقل فرصة مشاهدة بعض الألوان إن لم يستطيعوا مشاهدة الصور.

إننا نستطيع استخدام الصور بأحجام مختلفة طولياً أو عرضياً كخلفيات للصفحة، والمتصفح نفسه هو الذي يقوم تلقائياً بعرضها في وضع التجانب مما يعطي الانطباع بأنها صورة كبيرة.

ولنكمل مع باقي الخصائص في وسم <BODY>: ربما لاحظت خلال استخدامك للإنترنت أن معظم الوصلات التشعبية (Links) التي تنقر عليها لتنقلك إلى صفحات أو مواقع أخرى على الشبكة هي دائماً مميزة باللون الأزرق، وأن الوصلات التي قمت بزيارتها فعلاً قد تحول لونها إلى القرمزي. حسناً، هذه هي الألوان الإفتراضية التي تعتمدها المتصفحات. لكن قد لا يعجبك ذلك وتريد تغيير هذا النظام. أو ببساطة ربما تريد استخدام لون أو صورة غامقة لخلفية الصفحة بما سيؤدي إلى اختفاء هذه الوصلات أو حتى اختفاء نص الصفحة نفسها. فما العمل؟
إليك هذه الخصائص التي تقوم بالتحكم في ألوان النصوص:

TEXT="#rrggbb"	تحديد لون النص الأساسي للصفحة
LINK="#rrggbb"	تحديد لون الوصلات التشعبية
VLINK="#rrggbb"	تحديد لون الوصلات التشعبية التي تمت زيارتها visited links
ALINK="#rrggbb"	تحديد لون الوصلة التشعبية الفعالة أي عندما يتم النقر عليها active links

والآن، دعنا نجمل الخصائص السابقة في عبارة واحدة. وسوف أكتب الرموز الخاصة بالألوان بنفس تلك الألوان التي تمثلها. وألفت نظرك إلى أنه لا أهمية للترتيب في كتابة هذه الخصائص داخل العبارة.

```
<BODY BACKGROUND="backimag.jpg"
BGCOLOR="#ffff00"
TEXT="#000066"
LINK="#00ff00"
VLINK="#ff0000"
ALINK="#999999">
```

حاول أن تحللها! هل استنتجت أنني قد حددت الصورة backimag.jpg كخلفية للصفحة؟ وأنني اخترت اللون الأصفر للخلفية (في حالة عدم عرض الصورة السابقة كخلفية)؟ وان النص سيظهر باللون الأزرق الغامق؟ أما الوصلات التشعبية فلونها أخضر، والوصلات التي تمت زيارتها ستظهر باللون الأحمر. أما تلك الوصلة الفعالة فستظهر باللون الرمادي في لحظة النقر عليها بالفأرة.

إذا كانت هذه هي استنتاجاتك... فمبروك، لقد نجحت. وكل ما أتمناه أن تكون قد قضيت وقتاً ملوناً وزاهياً مع هذا الدرس.

الدرس الثالث
الخطوط

أهلاً وسهلاً بك إلى الدرس الثالث من دروس HTML. لا زلنا نناقش معاً أساسيات تنسيق صفحات الإنترنت والتحكم بخصائصها. وسوف نتابع ذلك في هذا الدرس من خلال التعرف على الوسوم الخاصة بالخطوط.
سوف تلاحظ في هذا الدرس والدروس اللاحقة أن هناك أكثر من طريقة لأداء نفس العمل، أو إعطاء نفس الخصائص لصفحات الإنترنت. وبالمقابل قد يبدو لك أن بعض الوسوم والخصائص متشابهة في تأثيرها، لكن بالقليل من التدقيق والتجربة ستكتشف أن لكل وسم خصوصيته.

ولنبدأ

راجع صفحتنا البسيطة التي عملنا فيها في الدرسين السابقين. إننا لم نقم بالتعامل مع الخطوط فيها ولا بأي شكل من الأشكال. أي أننا تركناها على إعداداتها الافتراضية.
وبالمناسبة فإن هذه الإعدادات هي خط عادي، نوعه Times New Roman وحجمه 3 (بمقياس متصفحات الإنترنت).

الوسم الأول الخاص بالخطوط هو <FONT> ... </FONT>
وهو يقوم بالتحكم بالخطوط من حيث النوع واللون والحجم. أما الخصائص التي نستخدمها مع هذا الوسم والوسوم الأخرى للخطوط فهي كالتالي:

Face	تقوم هذه الخاصية بتحديد نوع الخط الذي نريده، وقد نقوم بتحديد أكثر من نوع معاً. وفي هذه الحالة إذا لم يتواجد الخط المحدد أولاً على جهاز الشخص الذي يتصفح الموقع يتم إعتماد الخط الثاني ... وهكذا <FONT FACE="Traditional Arabic, Arabic Transparent, Simplified Arabic"> ... Text ... </FONT> طبعاً لا تنس أن تتأكد من كتابة أسماء الخطوط بالصورة الصحيحة هجائياً.
Color	أما هذه الخاصية فتحدد لون الخط، وذلك بنفس مبادئ تحديد الألوان التي تحدثنا عنها في الدرس السابق <FONT COLOR="#FF0000"> ... Text ... </FONT>
Size	ولتحديد حجم الخط نستخدم هذه الخاصية. وفقط هناك سبعة أحجام لأي خط تستطيع المتصفحات التعرف عليها. ونقوم بتحديد الحجم المطلوب بأسلوبين: أولهما المباشر. حيث يتم كتابة رقم يتراوح ما بين 1-7. أي أننا نختار الحجم الذي نريده مباشرة. <FONT SIZE="4"> ... Text ... </FONT>

وإليك نماذج بأحجام الخطوط

خط بحجم 1
خط بحجم 2
خط بحجم 3 (الخط الافتراضي)
خط بحجم 4
خط بحجم 5
خط بحجم 6
خط بحجم 7

أما الأسلوب الثاني فهو النسبي: حيث تكتب الأرقام من 1 إلى 6 مرفقة إما بإشارة + أو بإشارة −.

```
<FONT SIZE="+4">
... Text ...
</FONT>
```

وفي هذه الطريقة فإن الأرقام 1−6 تمثل درجات التكبير (+) أو التصغير (−) للخط وذلك نسبةً إلى الحجم الافتراضي. فمثلا الرقم +4 يعني تكبير الخط أربع درجات عن الحجم الافتراضي وهو 3، أي أنه يصبح بالحجم 7. بالمقابل فأن الرقم −1 يعني تصغير الخط درجة واحدة أي يصبح بالحجم 2.
ولتوضيح هذا الأسلوب، إليك هذه النماذج:

خط بحجم −3
خط بحجم −2
خط بحجم −1
خط بحجم +0 (أو −0 وهو الافتراضي)
خط بحجم +1
خط بحجم +2
خط بحجم +3
خط بحجم +4
خط بحجم +5

لاحظ أنه حتى في الأسلوب النسبي لا نستطيع الحصول على أكثر من سبعة أحجام للخطوط.

حتى وإن حاولنا كتابة أرقام أكبر أو أصغر كما فعلت هنا بكتابة الحجم −3 أو +5.	

والآن أعرف ماذا تريد أن تسأل، ستقول لقد ثبت حجم الخط على حده الأدنى عند الدرجة -2 وعلى حده الأعلى عند الدرجة +4. إذن ما الفائدة من وجود الدرجات الأخرى الأقل من -2 والأكبر من +4؟
حسنا وأنا أجيبك بسؤال آخر: ماذا لو قمنا بتغيير الحجم الافتراضي للخط في كل الصفحة إلى 1 بدلاً من 3؟ (وسوف نقوم بذلك فعلاً بعد قليل)، ألا نحتاج في هذه الحالة إلى الدرجات من +1 إلى +6 لتمثيل الأحجام الأكبر منه؟ وإذا قمنا بتحديد 7 كحجم إفتراضي ألا نحتاج إلى الدرجات من -1 إلى -6 لتمثيل الأحجام الأصغر منه؟ إذن نحن نحتاج فعلاً إلى هذه الدرجات لكي نغطي جميع الإحتمالات الواردة.
أرجو أن يكون هذا الجواب قد أقنعك :-)

وهذه بعض الأمثلة لتوضح لك كيفية استخدام هذا الوسم، وسوف أرفق نتيجة كل مثال بعده مباشرة.

```
<FONT FACE="arial" SIZE="6" COLOR="#FF0000">
This font is Arial, Size is 6, Color is Red
</FONT>
```

This font is Arial, Size is 6, Color is Red

```
<FONT FACE="arial" SIZE="+3" COLOR="#FF0000">
This font is Arial, Size is +3, Color is Red
</FONT>
```

This font is Arial, Size is +3, Color is Red

```
<FONT FACE="Times New Roman" SIZE="5" COLOR="#0000FF">
This font is Times New Roman, Size is 5, Color is Blue
</FONT>
```

This font is Times New Roman, Size is 5, Color is Blue

```
<FONT FACE="courier" SIZE="2" COLOR="#800000">
This font is Courier, Size is 2, Color is Maroon
</FONT>
```

This font is Courier, Size is 2, Color is Maroon

```
<FONT FACE="Arial" SIZE="5" COLOR="#00FF00"> This </FONT>
<FONT FACE="Times New Roman" SIZE="7" COLOR="#FF00FF"> is </FONT>
```

```
<FONT FACE="Arial" SIZE="2" COLOR="#FF0000"> multi </FONT>
<FONT FACE="Impact" SIZE="4" COLOR="#000000"> colors, </FONT>
<FONT FACE="Courier" SIZE="2" COLOR="#0000FF"> multi </FONT>
<FONT FACE="Times New Roman" SIZE="3" COLOR="#008080"> faces, </FONT>
<FONT FACE="Courier" SIZE="6" COLOR="#FFFF00"> and </FONT>
<FONT FACE="Arial" SIZE="5" COLOR="#808080"> multi </FONT>
<FONT FACE="Impact" SIZE="2" COLOR="#800000"> sizes </FONT>
<FONT FACE="Times New Roman" SIZE="7" COLOR="#00FFFF"> text </FONT>
```

This is multi **colors,** multi faces, and multi **sizes** text

---

```
<FONT FACE="Impact" SIZE="6" COLOR="#000000">C </FONT>
<FONT FACE="Impact" SIZE="6" COLOR="#008080">O</FONT>
<FONT FACE="Impact" SIZE="6" COLOR="#FF0000">L</FONT>
<FONT FACE="Impact" SIZE="6" COLOR="#0000FF">O</FONT>
<FONT FACE="Impact" SIZE="6" COLOR="#800000">R</FONT>
<FONT FACE="Impact" SIZE="6" COLOR="#FF00FF">S</FONT>
```

COLORS

ننتقل الآن إلى الوسم الثاني من الوسوم الخاصة بالخطوط وهو <BASEFONT>.
وعمله هو تحديد نوع الخط وخصائصه بالنسبة للصفحة كلها . أي أنه يقوم بتعريف نوع الخط الأساسي الذي سيستخدم في الصفحة من بدايتها إلى نهايتها ويحدد لونه وحجمه.
هل لاحظت انه وسم مفرد ولا يحتوي على وسم للنهاية؟ بالطبع ما الحاجة إلى وسم النهاية طالما أنه يتعامل مع الصفحة ككل ومع الإعدادات الأساسية لها، وليس مع كلمة أو سطر أو فقرة بذاتها. لذلك فإن هذا الوسم يكتب عادة في أول الملف، ويفضل مباشرة بعد وسم <BODY>. أما الخصائص المستخدمة معه فهي نفس الخصائص سالفة الذكر مع <FONT> ، (نستطيع استخدام الخاصية Name معه بدلاً من Face). وبنفس الطريقة وبدون أي اختلافات. وإليك هذه الشيفرة كمثال:

```
<BASEFONT Name="Arial" COLOR="#FF0000" SIZE="5">
```

وبدراسة هذا المثال نستنتج أنه يقوم بتعديل الخط الافتراضي للصفحة بحيث يصبح نوعه Arial وحجمه 5 ولونه أحمر.
وبالتالي فإن كل النصوص المكتوبة في تلك الصفحة سيطبق عليها هذا النمط من الخط. ما لم نقم طبعا باستخدام الوسوم <Font> ... </Font> لتعديلها والتحكم بمظهرها كما فعلنا في الأمثلة السابقة، فهي أكثر تحديداً وأكثر مرونة من الوسم <BASEFONT>

وبمناسبة الحديث عن الألوان وتغيير اللون الأساسي لنص الصفحة. ألا تذكر أننا في الدرس السابق تكلمنا عن الخاصية Text التي تكتب مع الوسم <Body> والتي استخدمناها لتحديد لون نص الصفحة... أنا لا زلت أذكر ذلك.
لا يوجد تعارض بين هذه الخاصية وخاصية Color في الوسم <BASEFONT> فأنت بكل بساطة تستطيع استخدام أي منهما في صفحتك. وإذا حدث واستخدمت كلاهما فإن اللون المحدد مع الوسم <BASEFONT> هو الذي سيطبقه المتصفح ويعتمده.

هناك وسوم خاصة تستخدم لتمييز العناوين Headings في صفحات الإنترنت وهي:

</Hn> ... <Hn>

وحرف n هو رقم بين 1-6 يمثل مستوى العنوان.

<H1> Heading 1 </H1>
<H2> Heading 2 </H2>
<H3> Heading 3 </H3>
<H4> Heading 4 </H4>
<H5> Heading 5 </H5>
<H6> Heading 6 </H6>

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

ونأتي الآن إلى التنسيقات والتأثيرات التي يمكن إضافتها إلى النصوص. وفيما يلي الوسوم الخاصة بها متبوعة بمثال ونتيجته:

* الخط الغامق (الأسود العريض)، ونستخدم له الوسوم التالية:

<B> ... </B>
<STRONG> ... </STRONG>

This is **Bold Text**	<B> Bold Text </B>
This is **Strong Text**	<STRONG> Strong Text </STRONG>

* الخط المائل

<I> ... </I>
<EM> ... </EM>

This is *Italic Text*	<I> Italic Text </I>
This is *Emphasized Text*	<EM> Emphasized Text </EM>

* الخط المسطر

<U> ... </U>

This is Undelined Text	<U> Undelined Text </U>

* الخط المرتفع

^{...}

This is Superscript Text	^{Superscript Text}

* الخط المنخفض

_{...}

This is Subscript Text	_{Subscript Text}

* خط كبير

<BIG> ... </BIG>

This is Big Text	<BIG> Big Text </BIG>

* خط صغير

<SMALL> ... </SMALL>

This is Small Text	<SMALL> Small Text </SMALL>

* نص يعترضه خط

<STRIKE> ... </STRIKE>
<S> ... </S>

This is ~~Striked Text~~	<STRIKE> Striked Text </SRTIKE>
This is ~~Striked Text~~ too	<S> Striked Text </S>

* نص الآلة الطابعة TeleType

<TT> ... </TT>

This is `TeleType Text`	<TT> TeleType Text </TT>

وهذا النص يعرف أيضاً بالنص موحَد المسافات Monospaced Text. ولتوضيح هذا المفهوم إليك المثال التالي:
إذا أخذنا الحرفين m,i وكتبنا كل منهما عشر مرات متتالية نلاحظ أن المساحة التي شغلها الحرف m هي أضعاف المساحة التي شغلها الحرف i

iiiiiiiiii
mmmmmmmmmm

أما عند استخدام الوسم <TT> ... </TT> فإن المساحة التي يشغلها كلا الحرفين تصبح موحدة

iiiiiiiiii
mmmmmmmmmmm

وهذه أمثلة تجمع بين عدة تنسيقات معاً:

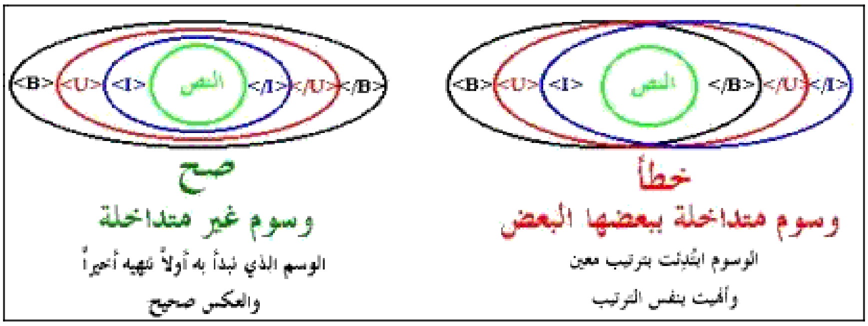
<B><I><U>
This is a Bold, Italic and Underlined Text
</U> </I> </B>

<u>This is a Bold, Italic and Underlined Text</u>

<FONT COLOR="#FF0000" SIZE="+3"><U><I>
This text is red, size +3, Italic, and Underlined
</I> </U> </FONT>

<u>This text is red, size +3, Italic, and Underlined</u>

وقد أردت من هذه الأمثلة توضيح مسائل معينة أولها: أن بإمكاننا استخدام عدة وسوم وتنسيقات معاً في نفس الوقت ولنفس المقطع من النص. (وذلك لجميع الوسوم وليس فقط لوسوم الخطوط). وكما ذكرت سابقاً، لا أهمية لترتيب هذه الوسوم ولا أيها ورد أولاً... **<u>لكن</u>**
عند استخدام الوسوم المتعددة في مقطع واحد يجب مراعاة عدم التداخل بينها!... كيف؟ أنظر إلى الرسم التالي:



فكتابة الوسوم السابقة بالطرق التالية هو خطأ:

<B><I><U>
This is a Bold, Italic and Underlined Text
</B> </I> </U>

<B><I><U>
This is a Bold, Italic and Underlined Text
</B> </U> </I>

أعرف أنك لم تصدقني وأنك قمت بتجربة هذه الوسوم وربما حصلت على نتيجة صحيحة. حسناً العبرة ليست في عبارة واحدة مكونة من وسمين أو ثلاثة تكتبها في ملف صغير بل في صفحة إنترنت كاملة قد تتألف من مئات أو حتى آلاف الوسوم مكتوبة في ملف خالٍ من الأخطاء المنطقية والتداخلات التي قد تسبب الإرباك للمتصفحات، وتؤدي إلى عدم عرض هذه الصفحة بالشكل المناسب والمطلوب.

لذلك فأهمية أن تتجنب وجود الوسوم المتداخلة في صفحتك هو بنفس الأهمية التي يجب أن توليها لكتابة هذه الوسوم بالصورة الصحيحة إملائياً. وإلا فالمتصفحات لا ترحم. وكثيرة هي المرات التي حصل فيها المصممون على صفحات منهارة بسبب نسيان حرف واحد أو إشارة مثل < أو > أو "

بإختصار شديد... وكقاعدة أساسية، الصفحة المصممة جيداً هي الصفحة ذات الوسوم الصحيحة وغير المتداخلة.

الدرس الرابع
الفقرات والقوائم

في هذا الدرس سوف نناقش الوسوم الخاصة بالفقرات بشكل خاص وترتيب الصفحات وتنسيقها بشكل عام.
صحيح أن استخدامك للألوان والرسومات في الصفحة يضفي عليها نوعاً من الحيوية، وأن الخطوط تعطي صفحتك رونقاً وجمالاً. لكنك إن لم تهتم بترتيب صفحتك أو تقضي بعض الوقت في تنسيق هيكلها العام وتنظيم فقراتها وقوائمها، فإنه من الصعب عليك الحصول على صفحة ويب ناجحة. فالترتيب هو الخطوة الأولى لجذب اهتمام الزائر أو القارئ لصفحتك وتسهل عليه فهم الخطوط العريضة للصفحة.

لقد قمت في الدرس الأول بإيضاح بعض الوسوم الخاصة بالفقرات. ولا بأس من تذكيرك بها. فالوسم <P> يقوم بإنهاء الفقرة. والوسم
 ينهي السطر الحالي وينقل النص إلى سطر جديد. والوسم   يقوم بإضافة الفراغات، ويجب تكرار كتابته بنفس عدد الفراغات المطلوب.
ونتابع في هذا الدرس مع هذه الوسوم وغيرها.

لقد قلت إن الوسم <P> هو وسم مفرد لكنه يستخدم أيضاً كوسم مزدوج <P> ... </P> وفي هذه الحالة يمكننا من تحديد إتجاه الفقرة وإتجاه النص فيها حيث يستخدم معه الخصائص ALIGN, DIR.

فالخاصية ALIGN تحدد محاذاة الفقرة وهي تأخذ القيم Left, Right, Center وأوضحها بالأمثلة التالية:

```
<P Align="left"> This is a left-aligned paragraph </P>
```

This is left-aligned paragraph

```
<P Align="right"> This is right-aligned paragraph</P>
```

This is a right-aligned paragraph

```
<P Align="center"> This is a centered paragraph</P>
```

This is a centered paragraph

كذلك لتوسيط الفقرات أو الكائنات بشكل عام في الصفحة نستطيع استخدام الوسوم <CENTER> ... </CENTER>

```
<CENTER> This is a centered text </CENTER>
```

This is a centered text

أما الخاصية DIR والتي نستخدمها أيضاً مع <P> فتقوم بتحديد إتجاه قراءة النص وتأخذ القيم

LTR	إتجاه النص من اليسار إلى اليمين (Left To Right)

RTL	إتجاه النص من اليمين إلى اليسار (Right To Left)

(تذكر هذه الخاصية جيداً فهي مهمة عند كتابة صفحات باللغة العربية)

ولتنسيق الفقرات أيضاً يوجد الوسوم
<BLOCKQUOTE> ... </BLOCKQUOTE> أي وسوم الفقرات المقتبسة. ووظيفتها تمييز الفقرة من خلال إدراج مسافة إضافية على الهامشين الأيمن والأيسر لها.

والحقيقة أنك تستطيع وضع عدة وسوم معاً إذا أردت إدراج هوامش أكبر. كما في المثال التالي:

```
<BLOCKQUOTE>
<BLOCKQUOTE>

النص هنا

</BLOCKQUOTE>
</BLOCKQUOTE>
```

وبالطبع ليس شرطاً أن تستخدم هذا الوسوم مع الفقرات المقتبسة فقط. فأنا مثلاً أضعها في بداية ونهاية كل صفحة من صفحات هذا الموقع. وبالتالي يظهر النص بعيداً قليلاً عن حاشية الصفحة فهذا أفضل من أن يكون ملاصقاً لها وأجمل.

والآن تأمل هذا الشكل وحاول أن تستنتج كيف قمت بإعداده...!؟

```
A          B          C          D
E          F          G          H
I          J          K          L
M          N          O          P
Q          R          S          T
```

ربما توصلت إلى أني استخدمت عدداً كبيراً من وسوم الفراغات   ونهاية السطر
. حسناً، إستنتاجك لا بأس به ولكنه ليس دقيقاً فأنا لم أستخدم أياً من هذه الوسوم هنا. بل كل ما فعلته بعد إعداد هذا الشكل هو وضعه ضمن:

<PRE> ... </PRE>

وهما اختصار لكلمة Preformated أي المنسق مسبقاً. وبالفعل فقد احتفظ هذا الشكل بالتنسيق المسبق الذي تم إعداده به. لكن تم تحويل الخط إلى خط موحد المسافات ولو لم أقم بوضعه ضمن هذه الوسوم لكانت النتيجة كالتالي:

A B C D E F G H I J K L M N O P Q R S T

لاحظ أن هذا الوسم يستخدم مع الفقرات التي لا نحتاج فيها إلى تنسيقات متعددة للخطوط أو الألوان. بل فقط مع الفقرات العادية موحدة الخط والتنسيقات.

القوائم

تحتوي لغة HTML على مجموعة من الوسوم الخاصة بتنظيم البيانات في قوائم وباستخدام عدة خيارات. وهناك نوعين من القوائم:
أولهما المتسلسلة Ordered Lists. واليك المثال التالي عليها

أسماء بعض المدن الفلسطينية

1. القدس
2. نابلس
3. رام الله
4. الخليل
5. جنين
6. طولكرم

وثانيهما القوائم غير المتسلسلة Unordered Lists وهذا مثال عليها

أسماء بعض الجامعات الفلسطينية

- جامعة النجاح
- جامعة القدس المفتوحة
- جامعة بيرزيت
- جامعة الخليل

عند التعامل مع القوائم بنوعيهما نحتاج إلى وسوم خاصة بتحديد بداية ونهاية القائمة ووسوم تحدد بنود هذه القائمة. بالنسبة للقوائم المتسلسلة نستخدم الوسوم

<OL> ... </OL>

أما بالنسبة للقوائم غير المتسلسلة فنستخدم

<UL> ... </UL>

ولتعيين كل بند من بنود القائمة نستخدم الوسم <LI> وهو وسم مفرد يكتب في بداية السطر الخاص بكل بند List Item.

إذن عندما قمت بإنشاء القوائم السابقة استخدمت الشيفرة التالية:

```
<OL>
<LI>القدس
<LI>نابلس
```

```
<LI>رام الله
<LI>الخليل
<LI>جنين
<LI>طولكرم
</OL>
```

```
<UL>
<LI>جامعة النجاح
<LI>جامعة القدس المفتوحة
<LI>جامعة بيرزيت
<LI>جامعة الخليل
</UL>
```

والخاصية الوحيدة التي تستخدم مع هذه الوسوم هي TYPE ووظيفتها تحديد شكل الرمز الظاهر مع بنود القائمة، وعادة تستخدم مع وسوم بداية القوائم <UL> أو <OL> وبذلك نحدد رمزاً واحداً لكل القائمة. ولكن نستطيع استخدامها أيضاً مع وسم البنود <LI> لإعطاء تحكم أكبر في مظهر القائمة من خلال تحديد رمز مختلف لكل بند.

فعند وضعها ضمن تعريف القوائم المتسلسلة تأخذ القيم: A, a, I, i التي تغير رموز الترقيم من الأرقام العادية الإفتراضية (والتي رمزها 1) إلى ترقيم باستخدام الأحرف اللاتينية الكبيرة أو الصغيرة، أو باستخدام الأرقام الرومانية كما ترى في الجدول التالي:

<OL TYPE="i">	<OL TYPE="I">	<OL TYPE="a">	<OL TYPE="A">
.i .ii .iii .iv .v	.I .II .III .IV .V	.a .b .c .d .e	.A .B .C .D .E

والحديث عن هذه الخاصية يقودني إلى الحديث عن مسألة مهمة في لغة HTML وهي مسألة الوسوم والخصائص المحددة بمتصفح معين دون غيره أي التي تعمل مع أحد المتصفحات ولا تعمل مع غيره.

والسبب في ذلك أن هذه الخاصية تستخدم أيضاً مع القوائم غير المتسلسلة، لكن ليس بصورة مطلقة...كيف؟ أنت ترى أن الرمز الموجود عند كل بند في القائمة هو عبارة عن نقطة سوداء يطلق عليها اسم Disc وهي المعرفة ضمناً في خاصية TYPE. لكن هناك رموز أخرى يمكن إظهارها وهي المربع square, والدائرة المفرغة circle وتعرف بالشكل التالي:

```
<UL TYPE="square">
<UL TYPE="circle">
```

ولكن للأسف هذه الخاصية لا تعمل ولا يظهر تأثيرها إلا مع متصفح نيتسكيب وليس مع مايكروسوفت إكسبلورر الذي يتعامل فقط مع القيمة الإفتراضية للخاصية. (رجاءً لا يغضب مستخدمو إكسبلورر فهناك الكثير من الوسوم والخصائص التي لا يستطيع نيتسكيب عرضها أيضاً).
ولإتمام الحديث عن القوائم، أذكر لك أن هناك وسوماً أخرى تستخدم لإنشاء القوائم غير المتسلسلة، وبنفس الطريقة المستخدمة مع <UL>...</UL> وهذه الوسوم هي:

<DIR> ... </DIR>
<MENU> ... </MENU>

هناك نوع خاص من القوائم يدعى قوائم الشرح أو التعريفات Definition Lists وكما يدل الإسم تستخدم عندما نريد إدراج قائمة من المصطلحات يتبع كل واحد منها شرح أو تعليق.

HTML
Hyper Text Markup Language
WWW
World Wide Web
FTP
File Transfer Protocol
GIF
Graphical Interchange Format
JPG, JPEG
Joint Photographic Experts Group

ونحتاج لإنشاء هذه القوائم إلى ثلاثة وسوم:
الأول <DL> ... </DL> لتعريف بداية ونهاية القائمة.
والثاني <DT> ويوضع قبل كل مصطلح لتحديده، وهو وسم مفرد.
أما الثالث فهو <DD> وهو وسم الشرح أو التعليق وهو أيضا مفرد. ولنقم الآن بكتابة شيفرة القائمة السابقة

```
<DL>
<DT>HTML <DD>Hyper Text Markup Language
<DT>WWW <DD>World Wide Web
<DT>FTP <DD>File Transport Protocol
<DT>GIF <DD>Graphical Interchange Format
<DT>JPG, JPEG <DD>Joint Photographic Experts Group
</DL>
```

الدرس الخامس
الصور والرسومات

في هذا الدرس سوف أقوم بالحديث عن الصور والرسومات وما يتعلق بالتعامل معها، بالإضافة إلى التعريف بأنواع الملفات الرسومية الدارجة في الإنترنت.

لقد اقتصر حديثنا عن الصور حتى الآن على إضافة خلفيات للصفحات، وكان ذلك في الدرس الثاني أما إدراج الصور ضمن الصفحات نفسها فله حكاية أخرى، أبدأ بروايتها لك الآن.

إن الوسم الرئيسي المستخدم لتعريف صورة ما داخل الصفحة هو <IMG> وهو وسم مفرد. لكن هل يكفي هذا لإدراج صورة؟ كلا، بالطبع يجب أن نحدد الصورة التي نريدها. لذلك نضيف الخاصية له SRC لتحديد موقع واسم الصورة.

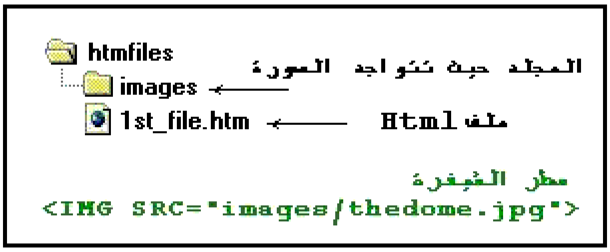
الصورة التالية إسمها thedome.jpg وعندما قمت بإدراجها. كانت الشيفرة الخاصة بذلك هي

```
<IMG SRC="thedome.jpg">
```



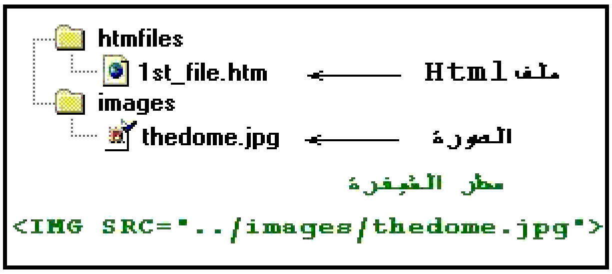
والصيغة هذه تفترض أن الصورة موجودة في ***نفس الدليل الفرعي أو المجلد*** حيث يتواجد ملف HTML الذي أعمل عليه، وقمت باستدعاء الصورة من خلاله. لكن ماذا لو كانت الصورة في مجلد فرعي آخر؟ حسنا سوف اناقش معك حالتين لهذه المسألة.

الحالة الأولى أن تكون الصورة موجودة في مجلد متفرع عن المجلد الموجود به ملف HTML حسب الشكل التالي:



نقوم في هذه الحالة بكتابة إسم هذا المجلد تتبعه إشارة / ثم اسم الصورة.

الحالة الثانية: أن يكون ملف HTML موجوداً في مجلد ما وتكون الصورة موجودة في مجلد آخر بنفس المستوى. أي أنهما مجلدين متجاورين وليسا متفرعين أحدهما عن الآخر.



وفي هذه الحالة نكتب .. (نقطتين) لتوجيه المتصفح للخروج من المجلد الفرعي الحالي (حيث يوجد ملف HTML) ومن ثم الدخول إلى المجلد images حيث توجد الصورة.

وبشكل عام، مهما كانت مواقع تواجد الملفات فإن عملية تحديد مواقعها والوصول إليها لا تخرج عن نطاق هذا النمط من الشيفرة. أي كتابة النقطتين للخروج من مجلد فرعي، وكتابة اسم المجلد الذي يجب الدخول إليه.

إن الأبعاد الأساسية لهذه الصورة هي 200×145 بيكسل Pixel وكما تلاحظ تم إدراج الصورة مع المحافظة على هذه الأبعاد. ومع ذلك فنحن نستطيع التحكم أيضاً بها وإظهار الصورة بالحجم الذي نريده من خلال هذا الوسم. كيف؟ بإضافة الخصائص HEIGHT, WIDTH متبوعة بأرقام تمثل الإرتفاع والعرض المطلوبين.

```
<IMG SRC="thedome.jpg" HEIGHT="70" WIDTH="120">
```



```
<IMG SRC="thedome.jpg" HEIGHT="300" WIDTH="500">
```



الخاصية التالية التي تستخدم مع <IMG> هي ALT وفيها نحدد نصاً بديلاً يظهر مكان الصورة. وهذا النص يلاحظ خصوصاً عندما يكون خيار "إظهار الصور تلقائياً" غير فعال في المتصفح. كما تستطيع ملاحظته في الفترة التي تسبق تحميل الصور وخاصة في المواقع بطيئة التحميل.

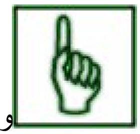
```
<IMG SRC="thedome.jpg" ALT="The Dome Of The Rock">
```

عندما نقوم بإدراج صورة ضمن فقرة فإن موقع ظهورها يتحدد بالطبع حسب ترتيب ورودها في الفقرة، مثلها مثل أي كلمة أو عبارة أخرى. ونستخدم الخاصية ALIGN لتحديد محاذاة الصورة مع النص المرافق لها أو لنقل بعبارة أخرى: ***تحديد موقع النص الذي يليها بالنسبة لها*** وهي تأخذ القيم: MIDDLE, LEFT, ,BOTTOM, TOP و RIGHT وأوضح لك تأثير كل قيمة كما يلي:

في الحالة العادية (مثل هذه) وعندما لا نقوم بتحديد أي محاذاة فإن النص الذي يلي الصورة يظهر بمحاذاة الحافة السفلى لها. وهذه هي الحالة الإفتراضية لظهور الصور والتي تمثلها القيمة BOTTOM

<IMG SRC="image.jpg" ALIGN="BOTTOM">

TOP



وعند تحديد هذه القيمة فإن السطر الأول من النص الذي يلي الصورة يقع بمحاذاة الحافة العليا لها. أما باقي النص فيمتد أسفلها.

<IMG SRC="image.jpg" ALIGN="TOP">

MIDDLE

أما عند تحديد هذه القيمة فإن السطر الأول من النص يقع بمحاذاة منتصف الصورة. كذلك فإن باقي النص يمتد أسفلها.

<IMG SRC="image.jpg" ALIGN="MIDDLE">

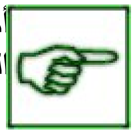
LEFT



هذه القيمة تؤدي إلى محاذاة الصورة إلى أقصى اليسار. مع التفاف النص الذي يليها على الجهة اليمنى ولعدة أسطر حسب ارتفاع الصورة.

<IMG SRC="image.jpg" ALIGN="LEFT">

RIGHT



أما هذه القيمة فتؤدي إلى محاذاة الصورة إلى أقصى اليمين. مع التفاف النص الذي يليها على الجهة اليسرى ولعدة أسطر حسب ارتفاع الصورة.

<IMG SRC="image.jpg" ALIGN="RIGHT">

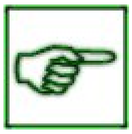
والآن بعد أن قمنا بتحديد محاذاة الصورة نحتاج إلى تحديد المسافة الفاصلة بينها وبين النص الذي يجاورها. ونستخدم لذلك الخصائص التالية:

VSPACE: لتحديد المسافة العمودية الفاصلة بين النص والحافتين العليا والسفلى للصورة.

HSPACE: لتحديد المسافة الأفقية الفاصلة بين النص والحافتين اليمنى واليسرى للصورة.

مثال:

<IMG SRC="image.jpg" ALIGN="RIGHT" VSPACE="20" HSPACE="20">



النتيجة: هذه الشيفرة ستدرج الصورة المسماه image.jpg مع محاذاتها ليمين الصفحة وإضافة مسافة فارغة مقدارها 20 بيكسل على الجهات الأربعة. (قارن بين هذا الإطار والإطار السابق الذي وضحت فيه خاصية ALIGN مع القيمة RIGHT. ولاحظ المسافة بين الصورة والنص المرافق لها.)

الخاصية الأخيرة والتي تستخدم مع الوسم <IMG> هي BORDER ووظيفتها إضافة إطار حول الصور والتحكم بسُمكِه. وهذه الخاصية تستخدم بشكل خاص عند تعيين صورة ما كوصلة تشعبية. ويتم التحكم بالسُمك من خلال إسناد رقم يمثل السُمك بالبيكسل. والقيمة الإفتراضية له هي 0 أي لا يوجد إطار حول الصورة.
مثلاً لإضافة إطار سُمكه 5 بيكسل نكتب الشيفرة التالية:

```
<IMG SRC="image.jpg" BORDER="5">
```

والآن حان الوقت لكي نناقش معاً بعض الأمور التي تتعلق بالصور والرسومات بشكل عام.

* هل حاولت أن تتعرف على أنواع الملفات الرسومية التي تقوم بتحميلها خلال تصفحك لمواقع الإنترنت؟
يزخر عالم الكمبيوتر بالعشرات من أنواع الملفات الرسومية وتنسيقات الصور. وكل منها يختلف عن غيره من عدة نواح، أذكر لك منها: الدقة، وعدد الألوان التي يستوعبها، والحجم التخزيني للملف. لكن هناك نوعين فقط من هذه الملفات يتم تداولهما حالياً في الإنترنت وهما:

JPEG ,JPG
إختصار لـِ Joint Photographic Experts Group. ويدعم هذا التنسيق صوراً بعيار 24 بت (أي 16.7 مليون لون). وميزة هذا التنسيق تتمثل في إمكانية ضغط الصور بنسب مختلفة عند تخزينها وبالتالي الحصول على صور صغيرة الحجم نسبياً.(أعني هنا حجم التخزين بالكيلوبايتات وليس أبعاد الصورة). لكن بالمقابل كلما إزدادت نسبة الضغط وصغر حجم الملف كان ذلك على حساب الجودة والوضوح.

GIF
إختصار لـِ Graphical Interchange Format وأقصى عدد للألوان في هذا التنسيق هو 265 لون. ومع ذلك فإن أحجام الصور المخزنة به كبير نسبياً مقارنة بتنسيق JPG. لكن هناك مزايا رائعة ينفرد بها تنسيق GIF مما يستدعي استخدامه في صفحات الويب، أولها القدرة على تخزين صور بخلفيات شفافة Transparent Images وثانيها الصور المتحركة Animated Gifs
والآن قد تسأل، أي من هذين التنسيقين أستخدم في صفحاتي؟! لا يوجد جواب قطعي لهذا السؤال لكن إليك هاتين المعادلتين:

JPG= الصور الحقيقية ذات العدد الكبير من الألوان، وذات الأبعاد الكبيرة
GIF= الصور قليلة الألوان وصغيرة الأبعاد مثل الأزرار.

---

* ما هي درجة إستبانة شاشتك Resolution؟ إذا كنت لا تعرف الجواب قم بفتح **لوحة التحكم** (control panel) في ويندوز وإختر أيقونة **العرض**(display) ثم اختر التبويب **إعدادات** (settings) وهناك سوف تشاهد **مساحة سطح المكتب** (screen resolution) الذي يدل على درجة إستبانة الشاشة، وعلى الأغلب ستكون 640×480 أو 800×600، وهناك درجات أعلى تعتمد على قدرة محول العرض. كذلك سوف تشاهد **لوح الألوان** (color quality) الذي يدل على عدد الألوان التي يمكن عرضها بالإعدادات الحالية للشاشة.
أما في ويندوز 3.11 أو 3.1 فاختر أيقونة برنامج إعداد Windows من لوحة التحكم فتظهر لك قائمة تجد بضمنها نوع وإستبانة الشاشة.

هذا الحديث يقودني إلى وحدة البيكسل Pixel وهي اختصار لـِ Picture Element. إذا كانت شاشتك بإستبانة 640×480 فهذا يعني أنها مقسمة(نظرياً) إلى شبكة من 640 عمود و480 سطر. *وبمنتهى البساطة، إن كل خلية من هذه الشبكة تمثل بيكسل* وبالطبع كلما زادت الإستبانة كلما صغر حجم وحدة البيكسل.

---

* هل سبق لك وأن سمعت بمصطلح Thumbnail ضمن مصطلحات الإنترنت؟ حسناً، لا تلتفت إلى الترجمة الحرفية لهذه الكلمة، والتي تعني "ظفر الإبهام". فالمقصود حقيقةً بها هي تلك الصورة الصغيرة جداً التي تقوم بالنقر عليها فتؤدي إلى عرض صورة بحجم أكبر. لذلك قد يكون المصطلح الأنسب لوصفها هو "العيّنة".
(وإذا كنت قد زرت أحد المواقع الإخبارية لرأيت كيف يتم عرض عينات وصور مصغرة للقطات الأحداث وعند النقر على العينة تظهر الصورة الأصلية. إذن أنت لست مجبراً على الإنتظار لوقت طويل لحين ظهور صورة ذات حجم كبير للقطة لست معنياً بها).

ومن الواضح أن استخدام العينات مفيد وعملي جداً وأن وضعها في المواقع التي تحتوي على العديد من الصور يؤدي إلى تقليل الزمن اللازم لتحميل الصفحات وتجنب ضياع الوقت بانتظار ظهور الصور الأصلية كبيرة الحجم. لأنها تعطي الزائر الحرية في النقر عليها إذا رغب في رؤية الأصل أو تجاهلها. أما كيف يتم عمل هذه العينات؟ فذلك باستخدام أحد برامج معالجة الرسوم كبرنامج Paint Shop Pro. بتصغير أبعاد الصور الأصلية إلى النسبة المطلوبة.

أعرف ماذا ستسأل الآن، ستقول ألم نتعلم قبل قليل كيفية عرض الصور مع التحكم بأبعادها؟ ألا يؤدي استخدام الخصائص WIDTH, HEIGHT إلى التحكم بحجم الصور وعرضها بنسب مصغرة حسب ما هو مطلوب؟
إن استخدامك لهذه الخصائص يؤدي إلى إظهار الصورة بحيث تبدو مصغرة، لكنك فعلياً قمت بإجبار متصفح الزائر على تحميل الصورة بالحجم والأبعاد الأصلية ثم عرضها بالحجم المصغر أي أنك في النهاية لم تحقق الغاية من وجود هذه العينات.

الدرس السادس
الوصلات التشعبية

Links... أو الوصلات التشعبية هي روح الإنترنت. وإذا كانت الإنترنت بمجملها هي شبكة العنكبوت فإن هذه الوصلات هي الخيوط التي تشكل هذه الشبكة وتؤلف حلقات الوصل بين الملايين من مواقعها. تنقر على وصلة ما فتنقلك إلى صفحة أخرى في نفس الموقع... وتنقر على وصلة أخرى لتنقلك كلياً إلى أحد المواقع في الجانب الآخر من العالم... وصلة تجعلك تحمّل ملفاً وأخرى تجعلك تشغّل مقطعا موسيقياً وثالثة تعرض لك صورة...
حسناً، من المؤكد أنك استنتجت الآن من هذه المقدمة أنك بصدد تعلم كيفية إدراج الوصلات التشعبية في صفحاتك... لقد صدق استنتاجك لذلك هيا إلى العمل...

هناك عدة خيارات للوصلات التشعبية، منها أن تكون الوصلة لموقع آخر، أو أن تكون لصفحة أخرى داخل الموقع نفسه، ومنها أن تكون لمكان آخر في نفس الصفحة (إلى أعلى أو أسفل على سبيل المثال) أو أن تكون وصلة لعنوان بريد إلكتروني E-mail وفي جميع الحالات فإن المبدأ واحد لكن تختلف بعض التفاصيل. وسوف أناقش معك كل حالة على حدة وبالتفصيل.

نستخدم الوسوم <A> ... </A> كوسوم أساسية لإدراج الوصلات التشعبية، وهي اختصار لكلمة Anchor. وهي لا تعمل لوحدها بل تتطلب إضافة خصائص معينة أولها وأهمها الخاصية **HREF** التي نحدد من خلالها الموقع الذي نريد الدلالة عليه، ويجب أن يكتب عنوان الموقع كاملاً.

<u>الحالة الأولى:</u> إدراج وصلة تشعبية تشير إلى موقع خارجي.
لنقم بإدراج وصلة تشعبية إلى أحد المواقع العربية الرائدة والرائعة، وهو موقع شركة صخر. وعنوانه http://www.sakhr.com في هذه الحالة يتم كتابة الشيفرة بالشكل التالي:

```
<A HREF="http://www.sakhr.com"> </A>
```

لكن بقي شيء واحد وهو العبارة أو الكلمة التي سيتم النقر عليها لتشغيل الوصلة، وهذه يجب أن توضع بين الوسمين <A> ... </A>. أي لكي تكتمل الوصلة السابقة يجب أن نكتب معها أي عبارة نريدها، لكي ينقر عليها الزائر فتنقله إلى العنوان المطلوب. ما رأيك بعبارة: Go To SAKHR والتي تصبح الشيفرة معها بالشكل التالي:

```
<A HREF="http://www.sakhr.com">Go To SAKHR</A>
```

وتظهر الوصلة كما يلي:

<u>Go To SAKHR</u>

لم تعجبك؟ ليس ذلك مشكلة فأنت تستطيع كتابة أي شيء تريده كعنوان للوصلة التي تريدها. ما رأيك لو جعلنا كلمة SAKHR هي فقط العنوان لهذه الوصلة.

```
Go To <A HREF="http://www.sakhr.com">SAKHR</A>
```

Go To <u>SAKHR</u>

بل إنك تستطيع إدراج صورة أو (زر) كبديل عن الكلمات –كما تشاهد في الكثير من المواقع– وكل ما عليك فعله في هذه الحالة هو كتابة الوسم الخاص بإدراج الصورة بين الوسمين <A> ... </A> بالشكل التالي:

```
<A HREF="http://www.sakhr.com"><IMG SRC="sakhrlgo.gif"></A>
```

والذي يؤدي إلى ظهور الصورة التالية كوصلة تشعبية لموقع صخر

وبشكل عام فإن أي شيء يوضع بين الوسمين <A> ... </A> سوف يكون الوسيلة أو العنوان الذي ينقلنا إلى الموقع المشار إليه في الوصلة التشعبية، سواءً كان هذا الشيء نصاً أو صورة أو كلاهما معاً.

والآن هل تلاحظ الإطار الظاهر حول الصورة؟ وهل تذكر متى قمنا بالحديث عن هذا النوع من الإطارات؟ نعم، في الدرس السابق. عند إدراج صورة كوصلة تشعبية يظهر حولها إطار سمكه 2 بيكسل وهذه هي الحالة الإفتراضية. وبالطبع نستطيع إزالته بكتابة الخاصية BORDER="0" ضمن وسم الصورة.

```
<A HREF="http://www.sakhr.com"><IMG SRC="sakhrlgo.gif" BORDER="0"></A>
```

أو حتى تكبيره بكتابة السمك المطلوب مع هذه الخاصية.

```
<A HREF="http://www.sakhr.com"><IMG SRC="sakhrlgo.gif" BORDER="6"></A>
```

ننتقل الآن إلى **<u>الحالة الثانية</u>**، وهي أن تشير الوصلة التشعبية إلى ملف موجود في نفس الموقع (أي ملف محلي) سواءً كان ملف HTML أو صورة أو غير ذلك. وفي هذه الحالة فإن ما يكتب مع الخاصية HREF هو اسم هذا الملف المطلوب الوصول إليه.

لنقم بإنشاء وصلة تشعبية تقودنا إلى الصفحة الرئيسية لهذا الموقع ولنفترض أن الملف الذي يحتويها اسمه index.html والشيفرة الخاصة بذلك هي:

```
<A HREF="index.html">Main Page</A>
```

Main Page

وأذكرك بأنك تستطيع إدراج صورة هنا أيضاً كعنوان للوصلة التشعبية وبنفس التفاصيل التي شرحتها في الحالة السابقة.

هيا ندرج صورة مصغرة كعنوان لوصلة تشعبية للصورة الأصلية.

```
<A HREF="nablus1.jpg"><IMG SRC="nablus_1.jpg" BORDER="0"></A>
```

في هذا المثال قمت بتحديد الصورة المصغرة المسماة nablus_1.jpg كوصلة تشعبية تصلنا إلى الصورة الأصلية المسماة nablus1.jpg



لكن إنتبه إذا كان الملف المطلوب والذي تريد الإشارة إليه موجوداً في مجلد مختلف عن المجلد الذي يوجد به الملف الحالي، فيجب عليك تحديد المسار الكامل لهذا الملف وذلك بنفس الطريقة التي ناقشناها في الدرس السابق عندما قمنا بإدراج الصور.

الحالة الثالثة هي أن نقوم بالإشارة إلى مكان آخر داخل نفس الصفحة، إلى أولها مثلاً أو إلى آخرها أو أي مكان آخر نريده...
طبعاً مهما بلغت درجة الذكاء والألمعية التي يتصف بها الكمبيوتر ومتصفح الإنترنت، فهما لا يستطيعان معرفة ما يدور بفكرك وبالتالي لا يستطيعان معرفة المكان الموجود في نفس الصفحة والذي تريد نقل زائرك إليه من خلال الوصلة التشعبية. لذلك يجب أن تقوم أنت بتحديده.

والمبدأ هنا هو أن تقوم بتعريف أو تسمية هذا المكان بإسم معين سوف تقوم لاحقا باستخدامه في الوصلة التشعبية. وفي هذه الحالة يتحتم عليك استخدام الخاصية الثانية للوسم <A> وهي NAME

لنقم معاً بإدراج وصلة تشعبية تقوم بنقل الزائر من مكان وجود الوصلة إلى فقرة أخرى
أول ما يجب فعله هو الذهاب إلى هذه الفقرة واختيار أول كلمة فيها ثم وضعها داخل الوسوم <A> ... </A>

```
<A>LINKS</A>
```

والآن حان الوقت لاستخدام الخاصية NAME فالخطوة الثانية هي تعريف هذه الكلمة بأي اسم نريده (المهم أن نبقى متذكرين له). سوف أقوم بإعطاء الاسم attrib1

```
<A NAME="attrib1">LINKS</A>
```

لقد أصبحت هذه الفقرة جاهزة لكي نقوم بإدراج وصلات تشعبية إليها من أي مكان في الملف، بل ومن أي ملف آخر ... وأكثر من ذلك أنه إذا أراد أحد ما في أحد المواقع الأخرى أن يضع وصلة تشعبية لها من موقعه فإن باستطاعته ذلك شرط أن يعرف الإسم الذي عرّفناها به وهذا ليس صعباً بالطبع.

الخطوة الثالثة هي إدراج الوصلة التشعبية لهذه الفقرة.
ويلزمنا هنا معرفة اسم الملف الذي توجد به هذه الفقرة (أي الملف الذي نعمل به) واسمه htutor06.html لأنه سيشكل المدخل الأساسي للوصول إلى الفقرة المحددة. وتكون شيفرة الوصول إلى هذه الفقرة هي كالتالي:

```
<A HREF="htutor06.html#attrib1">3rd Paragraph</A>
```

3rd Paragraph

لاحظ أننا لم نكتف بذكر اسم الفقرة لوحدها بل يجب أن تقرن باسم الملف الأب الذي يتضمنها من خلال إشارة #

أما **الحالة الأخيرة** والتي نقوم فيها بإدراج وصلة تشعبية لعنوان بريد إلكتروني، يؤدي النقر عليها إلى إطلاق برنامج البريد الإلكتروني للزائر بشكل تلقائي. فالإختلاف الوحيد الذي يطرأ هنا هو كتابة كلمة MAILTO بعد خاصية HREF لكي تدل على أن العنوان الذي يلي هو عنوان EMAIL وليس أي عنوان آخر

```
<A HREF="MAILTO:yahya@palnet.com"> Email Me </A>
```

Email Me

والآن بعد أن انتهيت من سرد أساسيات استخدام الوصلات التشعبية وإدراجها في صفحات الويب بقي هناك بعض التوضيحات والملاحظات التي أجد أن من المهم ذكرها لك.
عندما قمنا بالتوصيل إلى عنوان خارجي، سواء كان لموقع ويب أو عنوان Email لاحظنا أننا استخدمنا كلمات مفتاحية ميزت طبيعة هذا العنوان، وأعطت المتصفح فكرة عن طبيعة التعامل مع هذا العنوان وطريقة الاتصال به. فعندما أردنا التشعب إلى موقع الويب كتبنا كلمة HTTP والتي تدل على نوع البروتوكول المستخدم في الاتصال بهذا الموقع، وهو بروتوكول نقل النصوص المتشعبة HyperText Transfer Protocol وعندما كتبنا عنوان Email استخدمنا كلمة MAILTO قبل هذا العنوان. وبالتالي قمنا بالإيعاز للمتصفح بفتح برنامج البريد الإلكتروني الافتراضي وتجهيزه لإرسال رسالة إلى العنوان المدرج. وحتماً لقد صادفت مثل هذه الحالة كثيراً خلال تجولك في مواقع الويب.

لكن هنا مجالات أخرى لاستخدام الإنترنت ولكل منها بروتوكوله الخاص. فمثلاً هناك الآلاف من المزودات المنتشرة عبر الإنترنت والتي تحتوي على أعداد هائلة لا تحصى من الملفات والبرامج الجاهزة للتحميل ويتم الوصول إليها عبر بروتوكول خاص لنقل الملفات يدعى FTP (File Transfer Protocol). ومن هذه المجالات أيضاً والتي لا تقل أهمية عن الويب أو البريد الإلكتروني المجموعات الإخبارية News Groups أو مجموعات النقاش التي تختص كل منها بمناقشة موضوع معين. وهذه تعمل من خلال بروتوكول NNTP (Network News Transfer Protocol).

إن تعدد مجالات إستخدام الإنترنت وتعدد البروتوكولات فيها لا يعني أنك تحتاج لأن يكتظ سطح مكتبك بالعديد من البرامج للتعامل معها. فمعظم المتصفحات التي نستخدمها تحتوي على برامج خاصة تدعم هذه الخدمات.
فمثلاً عند النقر على عنوان مزود FTP يتم الدخول إليه مثله مثل أي موقع ويب عادي وتظهر قائمة المجلدات والملفات فيه بشكل مشابه للمستكشف في Windows. أما النقر على عنوان إحدى المجموعات الإخبارية فيؤدي إلى سلوك مشابه للنقر على عناوين البريد الإلكتروني، أي إطلاق برنامج تصفح خاص بالمجموعات الإخبارية يكون مدمجاً ضمن حزمة المتصفح الأصلي.

والآن... أعتقد أنه ليس من الصعب عليك استنتاج الكيفية التي نضيف بها وصلات تشعبية لمزود FTP.
إليك هذا العنوان لأحد المزودات التي يحتوي على الكثير من البرامج المجانية أو المشتركة

ftp://ftp.simtel.net/pub/simtelnet/win95/

وكل ما عليك فعله هو كتابة الشيفرة التالية:

```
<A HREF="ftp://ftp.simtel.net/pub/simtelnet/win95/">Simtel FTP Server</A>
```

Simtel FTP Server

أما بالنسبة للمجموعات الإخبارية فتكتب الوصلات التشعبية لها باستخدام الكلمة المفتاحية NEWS. فعلى سبيل المثال، لوضع وصلة تشعبية لمجموعة النقاش alt.html الخاصة بمناقشة لغة HTML نكتب الشيفرة التالية:

```
<A HREF="news:alt.html">Alt.Html</A>
```

Alt.Html

الدرس السابع
الجداول (1)

هذا الدرس سيكون الأول من درسين حول الجداول. وقد فضلت تجزئتها إلى قسمين وذلك لأهمية هذا الموضوع وتعدد خصائص الجداول واحتمالات استخدامها في صفحات الويب.

تعد الجداول من أقوى الأدوات التي تتضمنها لغة HTML وأكاد أجزم بأنه لا يوجد موقع في الإنترنت إلا ويستخدمها بصورة أو بأخرى. والحقيقة أن وضع الجداول في صفحات الويب لا يقتصر على تلك القوائم من البيانات التي نحتاج لترتيبها في صفوف وأعمدة، بل يتعدى ذلك إلى استخدامها في تصميم الصفحات نفسها وتنظيمها، والتحكم بمظهرها بصورة قوية وفعالة لا يمكن أداؤها مهما استخدمنا من وسوم خاصة بتنسيق الصفحات.

إن التعامل مع الجداول وإدراجها في صفحات الويب سهل جداً مثله مثل أي أداة من أدوات HTML لكنه قد يبدو لك مربكاً بعض الشيء وخاصة في البداية، وذلك لتعدد الخصائص التي تستعمل معها وتعدد الأوجه التي نستطيع التصرف بها. لكن لا تقلق فكل شيء يبدو صعباً في بدايته ولكن سرعان ما يصبح سهلاً.

هل أنت مستعد؟ إذن هيّا بنا...

بداية، إليك هذا الوصف البسيط للوسوم الأساسية الخاصة بالجداول

<TABLE>...</TABLE>	وسوم تعريف الجدول
<TR>...</TR>	Table Row وسوم تعريف الصف في الجدول
<TD> *Cell Data* </TD>	Table Data وسوم تعريف الخلايا في الصف وتعريف محتويات كل خلية

والآن لنتكلم بصورة أكثر دقة وتفصيلاً:

هذه هي الوسوم التي نبدأ بها لإدراج جدول مكون من خلية واحدة أو من مليون خلية... الأمر سيان

</TABLE> ... <TABLE>

والآن بعد إدراج هذين الوسمين، هناك سؤالين ينبغي الإجابة عليهما. الأول: كم عدد الصفوف التي نريدها في الجدول؟ ثلاثة، أربعة، مائة؟ لا بأس، قم بإضافة الوسوم

</TR> ... <TR>

بنفس عدد الصفوف التي تريدها. ولنفترض هنا أنها ثلاثة.

```
<TABLE>

<TR>
</TR>

<TR>
</TR>

<TR>
</TR>

</TABLE>
```

والسؤال الثاني هو، كم عدد الخلايا (أو الأعمدة) التي نريدها في كل صف؟
وهنا نضيف الوسوم

</TD> ... <TD>

بنفس عدد الخلايا المطلوب. ومن البديهي أن نكتبها بين الوسوم <TR> ... </TR> طالما أن الخلايا هي جزء من الصفوف. وهنا سأفترض أننا نريد خليتين في كل صف، وبذلك يجب تكرار كتابتها مرتين لكل صف.
وأذكرك أن النص الذي نريد إدراجه في الخلية يكتب ضمن هذين الوسمين.

```
<TABLE>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

</TABLE>
```

هل اتضحت لك الصورة الآن. أنظر إلى نتيجة العمل التي حصلنا عليها.

Data	Data
Data	Data
Data	Data

هناك شيء ما ينقص. بالطبع ... الحدود. انتظر قليلاً وستعرف الخاصية التي تقوم بإضافة الحدود للجدول وغيرها من الخصائص الأخرى. لأنني قبل أن أستمر أود أن ألفت نظرك لمسألة معينة في الجداول. وهي أن طريقة التعامل معها تتم على ثلاثة مستويات:

- مستوى الجدول ككل
- مستوى الصفوف ككل أو كل واحد على حده
- مستوى الخلايا ككل أو كل واحدة على حده.

ولكل من هذه المستويات خصائصه التي ينفرد بها كما أن هناك خصائص مشتركة تستخدم مع كل الوسوم.

نبدأ بمناقشة الخصائص التي تستخدم مع الوسوم <TABLE> ... </TABLE> وسأقوم أولاً بسردها لك، ومن ثم إدراج بعض الأمثلة التي توضحها.

BORDER	تقوم هذه الخاصية بإضافة حدود للجدول وتحديد سماكتها، والقيمة الإفتراضية لها هي صفر أي لا حدود <TABLE BORDER="5"> <TABLE BORDER="0">
WIDTH	نستخدم هذه الخاصية لتحديد عرض الجدول ككل. وهناك أسلوبين لتحديد العرض: المطلق أي بكتابة الرقم الذي يمثل العرض بصورة مباشرة. والنسبي أي كتابة رقم نسبي مئوي يحدد عرض الجدول حسب عرض نافذة المتصفح. (أي أن عرض الجدول سيختلف باختلاف عرض نافذة المتصفح). <TABLE WIDTH="600"> <TABLE WIDTH="80%">
HEIGHT	لتحديد ارتفاع الجدول، ويكون تحديد هذا الإرتفاع من خلال قيمة مطلقة تحدد الإرتفاع بالبيكسل. أو قيمة نسبية تحدد ارتفاع الجدول بالنسبة لإرتفاع صفحة المتصفح <TABLE HEIGHT="500"> <TABLE HEIGHT="100%">
CELLSPACING	لتحديد المسافة بين كل خلية من خلايا الجدول <TABLE CELLSPACING="10">
CELLPADDING	لتحديد المسافة الفاصلة بين الحدود وبداية النص في كل خلية. أو لنقل: تحديد حجم الهوامش لخلايا الجدول. <TABLE CELLPADDING="10">

ALIGN	لتحديد محاذاة الجدول أفقياً على الصفحة يميناً أو يساراً. وهو يأخذ القيم left ,right <TABLE ALIGN="Left"> <TABLE ALIGN="Right">
BGCOLOR	ويستخدم لتحديد لون الخلفية للجدول <TABLE BGCOLOR="#00FFFF">

هذه هي الخصائص المستعملة مع الجدول. وسأقوم الآن بتطبيقها على المثال الوارد في بداية هذا الدرس وسأكتفي بكتابة وسم البداية أما باقي الوسوم فهي نفسها:

<TABLE BORDER="5">

Data	Data
Data	Data
Data	Data

<TABLE BORDER="5" CELLPADDING="5">

Data	Data
Data	Data
Data	Data

<TABLE BORDER="5" CELLPADDING="5" CELLSPACING="10">

Data	Data
Data	Data
Data	Data

<TABLE BORDER="5" CELLPADDING="5" CELLSPACING="10" BGCOLOR="#FFFF00">

Data	Data
Data	Data
Data	Data

<TABLE BORDER="5" CELLPADDING="5" CELLSPACING="10" BGCOLOR="#FFFF00" HEIGHT="300">

Data	Data
Data	Data
Data	Data

<TABLE BORDER="5" CELLPADDING="5" CELLSPACING="10" BGCOLOR="#FFFF00" HEIGHT="300" WIDTH="75%">

Data	Data
Data	Data
Data	Data

ونتكلم الآن عن الخصائص المستخدمة مع وسوم الصف <TR> ... </TR> ولا بأس من تذكيرك أن عدد الصفوف في الجدول يتحدد بعدد هذه الوسوم. أما أهم الخصائص التي تضاف لهذا الوسم فهي:

ALIGN	لتحديد محاذاة النص أفقياً داخل الخلايا التي يتكون منها الصف، والقيم المحتملة لها هي Right, Left, Center والقيمة الإفتراضية هي Center
VALIGN	لتحديد المحاذاة العمودية للنص داخل خلايا الصف، وذلك إما للأعلى أو للأسفل أو في المنتصف أو على امتداد الخط الأساسي للخلية. وقيمها على التوالي هي: Baseline ,Top, Bottom, Middle
BGCOLOR	لتحديد لون الخلفية للخلايا التي يتكون منها الصف. وهنا يتم تجاهل لون الخلفية المحدد ضمن وسم <TABLE> ويتم تطبيق اللون المحدد هنا.

ونعود الآن إلى جدولنا السابق لنطبق عليه هذه الخصائص من خلال الأمثلة التالية:

```
<TABLE BORDER="5" HEIGHT="300">
<TR ALIGN="Left">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR ALIGN="Right">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR ALIGN="Center">
<TD> Data </TD>
<TD> Data </TD>
</TR>
</TABLE>
```

Data	Data
Data	Data
Data	Data

```
<TABLE BORDER="5" HEIGHT="300">
<TR VALIGN="Top">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR VALIGN="Bottom">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR VALIGN="Baseline">
<TD> Data </TD>
<TD> Data </TD>
</TR>
</TABLE>
```

Data	Data
Data	Data
Data	Data

```
<TABLE BORDER="5" HEIGHT="300" BGCOLOR="#FFFFFF">
<TR BGCOLOR="#808080">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR BGCOLOR="#C0C0C0">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR>
<TD> Data </TD>
<TD> Data </TD>
</TR>
</TABLE>
```

Data	Data
Data	Data
Data	Data

```
<TABLE BORDER="0" HEIGHT="100%" WIDTH="100%">
<TR BGCOLOR="#808080">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR BGCOLOR="#C0C0C0">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR BGCOLOR="#FFFFFF">
<TD> Data </TD>
<TD> Data </TD>
</TR>
</TABLE>
```

Data	Data
Data	Data
Data	Data

والآن ماذا بقي لدينا؟ بالطبع بقيت الوسوم الخاصة بالخلايا. وسوف أتابع الحديث عنها في الدرس القادم إن شاء الله.

الدرس الثامن
الجداول (2)

نتابع معاً في هذا الدرس الحديث عن الجداول. وأنا أفترض أنك قد أنهيت الدرس السابق بنجاح، وأن لديك الآن فكرة جيدة جداً عن الجداول وكيفية إنشائها والتعامل مع خصائصها ومع الصفوف وخصائصها. ونكمل الآن من حيث توقفنا، أي مع خصائص الخلايا.

هل تذكر ما قلناه عن عدد الخلايا في الصف الواحد؟ إن عدد الخلايا المطلوب يتحدد من خلال كتابة الوسوم <TD> ... </TD> مرات بنفس العدد المطلوب. ومن الممكن أن تحتوي الخلية على أي عنصر من عناصر لغة HTML : نصوص، رسوم، قوائم، وصلات تشعبية، بل وحتى جداول. (نعم، تستطيع إدراج جدول داخل جدول آخر)

لنسترجع معا المثال الذي قمنا بالتدرب عليه في الدرس السابق، فسوف نكمل هذا الدرس معه. وهو جدول صغير مكون من ثلاثة صفوف وعمودين (أي خليتين في كل صف).

```
<TABLE>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

</TABLE>
```

أما الخصائص المستخدمة مع الخلايا، فهذا جدول بها:

ALIGN	تحدد محاذاة النص الموجود في الخلية أفقياً، والقيم المستخدمة هي Left, Center, Right
VALIGN	تحدد المحاذاة العمودية للنص، وهو يأخذ القيم Top, Middle, Bottom, Baseline

WIDTH	تحدد عرض الخلية، وذلك بكتابة القيمة المباشرة للعرض المطلوب بالبيكسل، أو بكتابة رقم يمثل النسبة المئوية. ويكفي تحديد العرض للخلايا في أحد الصفوف لكي يتم تطبيقه على كل الخلايا في كل الصفوف.
HEIGHT	تحدد الإرتفاع المطلوب للخلية في الصف، وذلك بالطرق المباشرة أو النسبية. وقيامك بتحديد ارتفاع إحدى الخلايا في الصف يؤدي إلى تطبيقه على كل الخلايا فيه.
BGCOLOR	تحدد لون خلفية الخلية
COLSPAN	يقوم بدمج الخلية الحالية مع العدد المطلوب من الخلايا التي تليها أفقياً <TD COLSPAN="n"> حيث n هو عدد الخلايا التي سيتم دمجها
ROWSPAN	يقوم بدمج الخلية الحالية مع العدد المطلوب من الخلايا التي تليها عمودياً (أي أسفلها). <TD ROWSPAN="n"> وبالطبع n هو عدد الخلايا التي سيتم دمجها

وقبل أن نستمر، يبدو لي أن هناك بعض الملاحظات المهمة التي ينبغي ذكرها:

- كما تلاحظ هناك خصائص تتكرر مع جميع الوسوم. خذ مثلاً الخاصية BGCOLOR. كيف يتم التعامل معها إذا كررت مع جميع الوسوم؟ بكل بساطة يتم تطبيق اللون المحدد مع وسم الخلية، فإذا لم يكن محدداً يطبق اللون المحدد مع وسم الصف، فإذا لم يوجد يطبق اللون المحدد مع وسم الجدول. وإذا لم يكن هذا محدداً بدوره يتم إعتماد لون خلفية الصفحة المحدد في الوسم <BODY>.
- الملاحظة الثانية تتعلق بالخصائص WIDTH, HEIGHT. يختلف أسلوب التعامل مع هذه الخصائص من متصفح لآخر، بل وتختلف أيضاً طريقة تفسير القيم المحددة معها وخصوصاً فيما يتعلق بالنسب المئوية. (راجع الموضوع: الوسوم الخاصة والمتصفحات).
وبدون الخوض في تفاصيل هذه الإختلافات التي لن تؤدي إلا إلى المزيد من الإشكالات لديك... وبعد التجربة يبدو أن أفضل طريقة للتعامل مع هذه الخصائص هي قيامك بتحديد العرض (وكذلك الإرتفاع إذا أردت ذلك) للجدول ككل من خلال الوسم <TABLE>. ثم استخدام هذه الخصائص في وسوم الخلايا وتحديد العرض المطلوب لكل خلية على حده في الصف الأول، والارتفاع المطلوب لكل صف في الجدول.
وهذه برأيي أفضل طريقة تضمن بها أفضل مشاهدة للجدول لجميع زوار موقعك.
- إذا أردت أن تحتوي بعض الصفوف في الجدول على عدد من الخلايا أقل من باقي الصفوف، فلا يكفي أن تقوم بحذف وسوم الخلايا منها. (كما ترى في الشيفرة التالية:)

```
<TABLE BORDER="5">

    <TR>
```

```
                <TD> Data </TD>

        </TR>

        <TR>

                <TD> Data </TD>
                <TD> Data </TD>

        </TR>

        <TR>

                <TD> Data </TD>

        </TR>

</TABLE>
```

لأن هذا ما ستحصل عليه:

	Data
Data	Data
	Data

لقد بقي مكان الخلايا المحذوفة محجوزاً كما لو أنها لم تحذف. أما الخلايا الباقية فظلت محتفظة بنفس خصائصها، أي أننا لم نستفد من عملية الحذف. والحقيقة أن الطريقة المثلى لذلك هي أن تقوم ***بدمج*** الخلايا معاً وذلك باستخدام الخصائص COLSPAN, ROWSPAN.

إذن لنقم بإعادة كتابة شيفرة الجدول مع استخدام هذه الخصائص:

```
<TABLE BORDER="5">

<TR>

<TD COLSPAN="2"> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>
```

```
</TR>

<TR>

<TD COLSPAN="2"> Data </TD>

</TR>

</TABLE>
```

Data	
Data	Data
Data	

لاحظ أن العدد 2 هو عدد الخلايا التي قمنا بدمجها. ولاحظ أيضاً انني لم أقم بإعادة وسوم الخلايا المحذوفة لأننا أصلاً لا نحتاج لها بعد أن قمنا بالدمج. **وكقاعدة أساسية: كل خلية يتم دمجها يجب بالمقابل حذف وسوم التعريف الخاصة بها. ما عدا تعريف الخلية الأساسية بالطبع.**

مثال آخر: لنقم بدمج الخلايا الموجودة في العمود الأول

```
<TABLE BORDER="5">

<TR>

<TD ROWSPAN="3"> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>

</TR>

</TABLE>
```

ومرة أخرى بعد تعريف خاصية الدمج العمودي، قمت بحذف تعريف الخلايا المدموجة من الصف الثاني والثالث. وهذا هو الجدول الناتج.

Data	Data
Data	

Data	

هناك نوع خاص من الخلايا التي يتم تعريفها باستخدام الوسوم <TH> ... </TH> وهي اختصار Table Header أي ترويسة الجدول.
والفرق الوحيد بينها وبين <TD> ... </TD> هو أن النص الذي تحتويه يظهر بخط أسود عريض ومحاذاته في منتصف الخلية بصورة إفتراضية. (ليس بالشيء المهم، كما أعتقد)، خاصة وأن الخصائص المستخدمة معها هي نفس خصائص <TD> وبنفس التفاصيل التي ذكرت.

الوسوم الأخيرة المستخدمة في الجداول هي <CAPTION> ... </CAPTION> وهي تختص بإضافة عنوان رئيسي للجدول ككل. لذلك فهي عندما تكتب يتم وضعها مباشرة بعد الوسم <TABLE> وبصورة مستقلة وليس ضمن وسوم الصفوف أو الخلايا.

```
<TABLE BORDER="5">
  <CAPTION> Table Caption </CAPTION>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

</TABLE>
```

Table Caption

Data	Data
Data	Data
Data	Data

وأخيراً... وصلنا إلى نهاية هذه الدرس. وما بقي لدي هو أن أحثــك على تطبيق ما قمت بشرحه فيه وفي الدرس السابق وباقي الدروس، وتجربة جميع الاحتمالات الواردة للخصائص والقيم. إليك عدة طرق وأمثلة تطبيقية لاستخدام الجداول.

أمثلة تطبيقية
على الجداول

```
<TABLE WIDTH="80%" BORDER=10 BGCOLOR="#C0C0C0">
   <TR>
       <TD> ... </TD>
   </TR>
</TABLE>
```

جدول مكون من خلية واحدة

```
<TABLE WIDTH="80%" BGCOLOR="#C0C0C0">
   <TR>
       <TD HEIGHT="200"> ... </TD>
   </TR>
</TABLE>
```

جدول مكون من خلية واحدة وبدون حدود

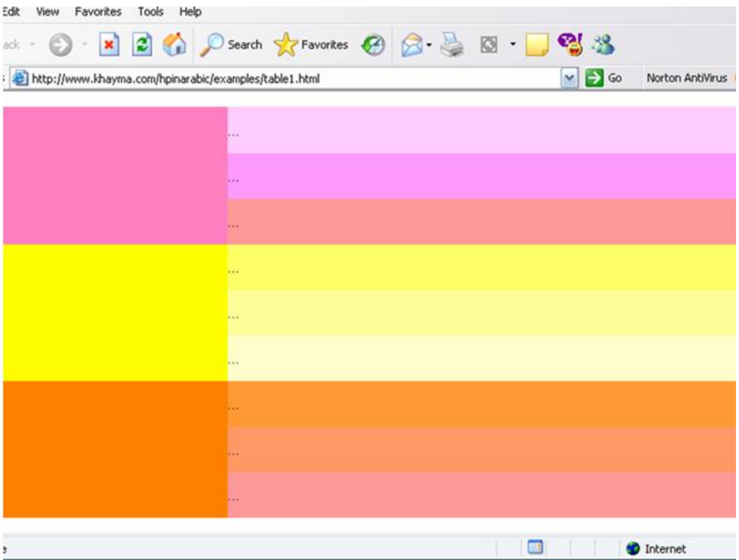
```
<TABLE BORDER="10">
   <TR>
      <TD><IMG SRC="image.jpg"></TD>
   </TR>
</TABLE>
```




```
<TABLE CELLPADDING="5" CELLSPACING="5" WIDTH="50%" BORDER="3" BGCOLOR="#FFFF00">
   <TR>
      <TD>...</TD>
      <TD>...</TD>
      <TD>...</TD>
   </TR>
   <TR>
      <TD>...</TD>
      <TD>...</TD>
      <TD>...</TD>
   </TR>
   <TR>
      <TD>...</TD>
      <TD>...</TD>
      <TD>...</TD>
   </TR>
   <TR>
      <TD>...</TD>
      <TD>...</TD>
      <TD>...</TD>
   </TR>
</TABLE>
```

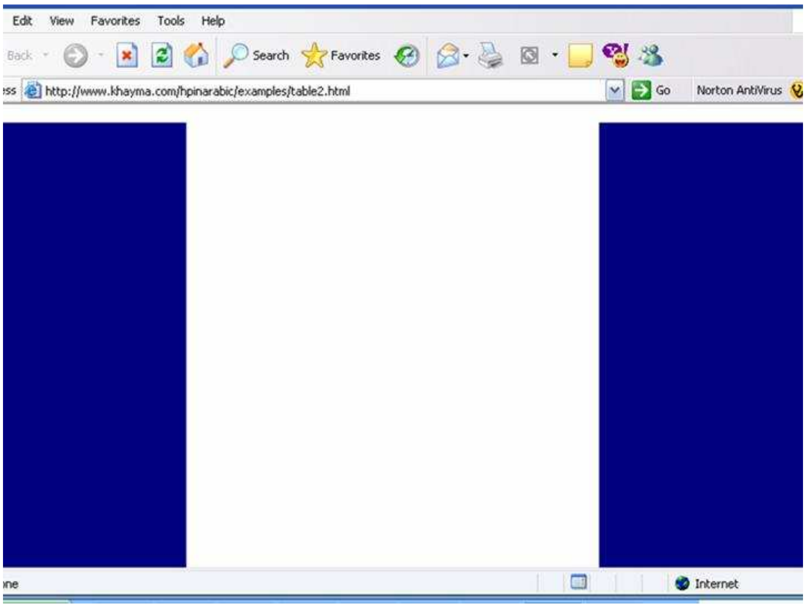
...	...	...
...	...	...
...	...	...
...	...	...

```
<TABLE CELLPADDING=0 CELLSPACING=0 WIDTH="100%" HEIGHT="100%" BORDER=0>
  <TR>
    <TD WIDTH="33%" ROWSPAN=3 BGCOLOR="#FF80C0">...</TD>
    <TD BGCOLOR="#ffCCFF">...</TD>
  </TR>
  <TR>
    <TD BGCOLOR="#ff99FF">...</TD>
  </TR>
  <TR>
    <TD BGCOLOR="#ff9999">...</TD>
  </TR>
  <TR>
    <TD ROWSPAN="3" BGCOLOR="#FFFF00">...</TD>
    <TD BGCOLOR="#ffff66">...</TD>
  </TR>
  <TR>
    <TD BGCOLOR="#ffff99">...</TD>
  </TR>
  <TR>
    <TD BGCOLOR="#ffffcc">...</TD>
  </TR>
  <TR>
    <TD ROWSPAN="3" BGCOLOR="#FF8000">...</TD>
    <TD BGCOLOR="#ff9933">...</TD>
  </TR>
  <TR>
    <TD BGCOLOR="#ff9966">...</TD>
  </TR>
  <TR>
    <TD BGCOLOR="#ff9999">...</TD>
  </TR>
</TABLE>
```

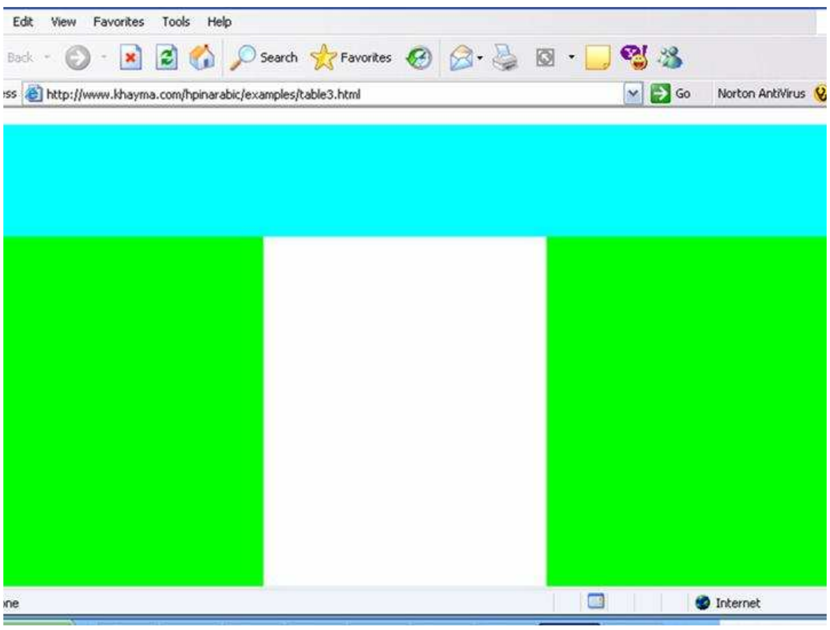



```
<TABLE WIDTH="100%" BGCOLOR="#FFFFFF" BORDER=0 height="2000">
  <TR>
    <TD WIDTH="25%" BGCOLOR="#000080">...</TD>
    <TD WIDTH="50%">...</TD>
    <TD WIDTH="25%" BGCOLOR="#000080">...</TD>
```

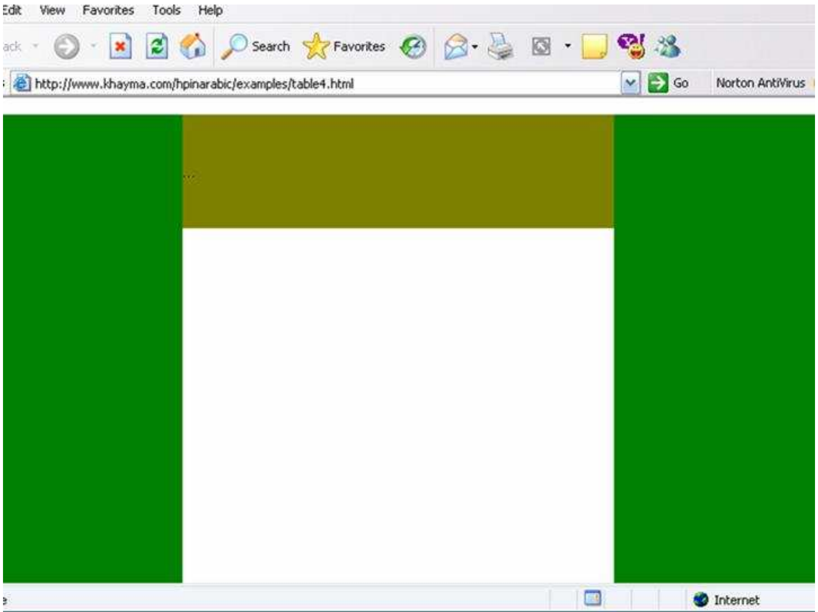
```
    </TR>
</TABLE>
```




```
<TABLE WIDTH="100%" BORDER=0 CELLSPACING=0>
    <TR>
        <TD COLSPAN=3 BGCOLOR="#00FFFF" HEIGHT="100">...</TD>
    </TR>
    <TR>
        <TD BGCOLOR="#00FF00" HEIGHT="2000">...</TD>
        <TD BGCOLOR="#FFFFFF">...</TD>
        <TD BGCOLOR="#00FF00">...</TD>
    </TR>
</TABLE>
```



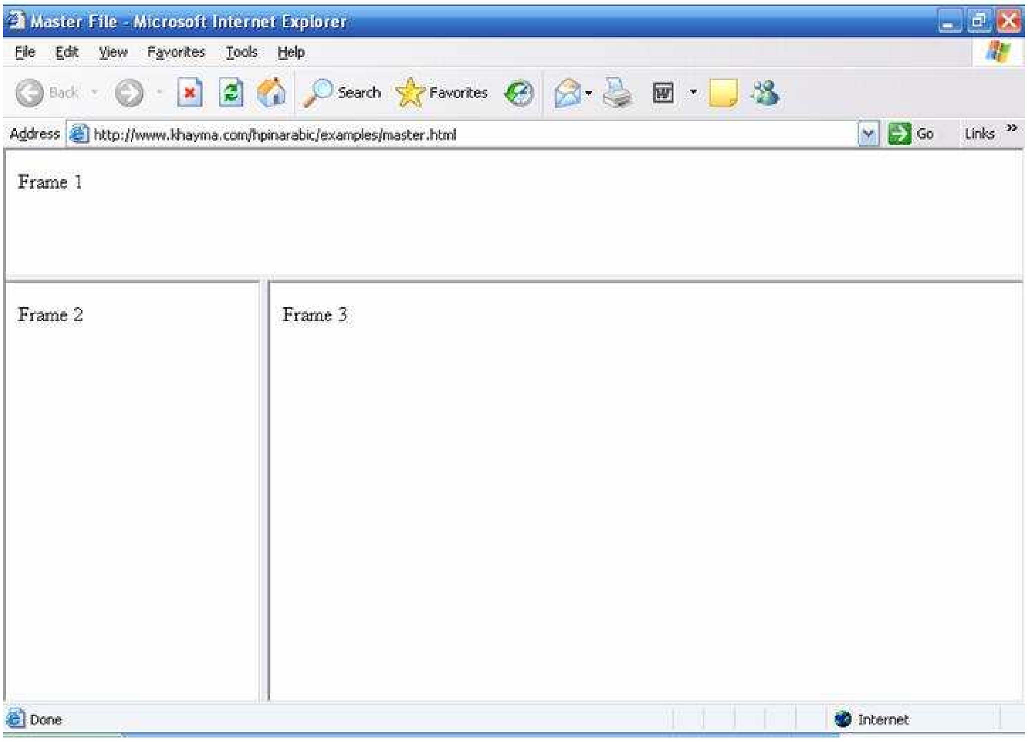
```
<TABLE CELLPADDING=0 CELLSPACING=0 WIDTH="100%" BORDER=0>
  <TR>
    <TD WIDTH="25%" ROWSPAN=2 BGCOLOR="#008000">...</TD>
    <TD BGCOLOR="#808000" HEIGHT="100">...</TD>
    <TD WIDTH="25%" ROWSPAN=2 BGCOLOR="#008000">...</TD>
  </TR>
  <TR>
    <TD HEIGHT="2000">...</TD>
  </TR>
</TABLE>
```



الدرس التاسع
الإطارات (1)

في هذا الدرس والدرسين التاليين سوف نقوم بالتعرف على الإطارات Frames وطريقة عرض صفحات الويب باستخدامها...

فهل تعرف ما هي الإطارات؟ حسنا، سأوضحها لك... هل سبق لك وأن زرت إحدى الصفحات لتشاهد أنها مقسمة إلى عدة أقسام بحيث يظهر في كل منها صفحة مستقلة، وتبدو بصورة منفصلة عن الأقسام الأخرى. وربما تكون قد قمت بالنقر على إحدى الوصلات التشعبية الموجودة في أحد الأقسام لتظهر الصفحة المتعلقة بها في القسم الآخر.
إذا كنت قد شاهدت مثل هذه الصفحات فهذا يعني أن الإطارات مألوفة لديك وإلا فشاهد مثالاً على صفحة ذات إطارات



كما تشاهد، فإن الصفحة مكونة من ثلاثة أقسام: علوي وأيسر وأيمن... والحقيقة أن كل قسم منها هو عبارة عن ملف Html كامل ومستقل بحد ذاته. وهي مجرد صفحات عادية لا تختلف أبداً عن تلك التي تعلمت إنشاءها في الدروس السابقة، ولا علاقة لكل منها بالصفحات الأخرى من حيث التركيب والتعريف.
أما كيف تم جمعها معا لتظهر بالشكل الذي تشاهده؟ فهنا بيت القصيد. فبالإضافة إلى الصفحات والملفات الإعتيادية يوجد دائماً ملف أساسي يتم إنشاؤه خصيصا لتعريف صفحة الإطارات وتجميعها وتحديد خصائصها. أي أن المعادلة تتلخص بـِ:

مكونات صفحة الإطارات = عدد ملفات الصفحة نفسها + صفحة الملف الأساسي الذي يجمعها.

أي أنني في المثال السابق إحتجت فعلياً إلى أربعة ملفات لتكوين الصفحة.

وقبل أن نبدأ... لنقم بالتحضير للأمثلة التي سترد في هذا الدرس. لذلك قم بإنشاء ثلاثة أو أربعة ملفات بسيطة لكي تستخدمها في تطبيق الأمثلة أو استخدم ملفاتك القديمة التي قمت بالتدرب عليها في الدروس السابقة. أنا قمت بإنشاء ملفات على النمط التالي (وهي التي استخدمتها في المثال) وأسميتها frame1.html, frame2.html, frame3.html

```
<HTML>
<HEAD>
<TITLE>Frame1</TITLE>
</HEAD>
<BODY>
Frame 1
</BODY>
</HTML>
```

ونبدأ الآن بتعريف الملف الرئيسي الذي سيضم كافة الإطارات والملفات. وهو بالمناسبة ملف ذو حالة خاصة حيث نقوم باستخدام الوسوم

<FRAMESET> ... </FRAMESET>

بدلاً من الوسوم <BODY> ... </BODY>

((إذن الملف الرئيسي للإطارات لا يتضمن تعريفا باستخدام BODY))

```
<HTML>
<HEAD>
<TITLE>Master File</TITLE>
</HEAD>

<FRAMESET>
</FRAMESET>

</HTML>
```

نأتي الآن إلى الخصائص: والخاصية الأولى التي تستخدم مع هذه الوسوم هي COLS وهي تعرّف عدد وأحجام الإطارات العمودية للصفحة. وتُحدد الأحجام بطريقتين (هل عرفتهما؟) نعم... إنهما الطريقة المباشرة والطريقة النسبية...أو كلاهما معا.

والآن إليك هذه الأمثلة التي توضح مفهوم الأعمدة... **لكن إنتبه! فهذه الشيفرة غير مكتملة وكتابتها بهذا الشكل فقط لن يؤدي إلى أي نتيجة ولا إلى ظهور أي إطارات** حيث ينقصها وسوم أخرى خاصة بمصدر الملفات الظاهرة داخل الإطارات، وسأقوم بسرد باقي الخصائص المهمة لاحقاً.

<FRAMESET COLS="50%,50%"> </FRAMESET>	يحدد إطارين عموديين حجم كل منهما 50% من حجم الشاشة
<FRAMESET COLS="20%,50%,30%"> </FRAMESET>	يحدد ثلاثة إطارات أحجامها 20% و 50% و 30% على التوالي من حجم الشاشة
<FRAMESET COLS="200,300,*"> </FRAMESET>	يحدد ثلاثة إطارات عمودية الأول حجمه 200 بيكسل، والثاني 300 بيكسل،

	أما الثالث * أي انه غير محدد بحجم معين ولكنه سيكون بالحجم المتبقي من الشاشة (طالما أننا لا نعرف استبانة الشاشة التي يستخدمها زائر الموقع)
<FRAMESET COLS="200,*,15%,20%"> </FRAMESET>	يحدد أربعة إطارات حجم الأول هو 200 بيكسل، والثالث 15% من حجم الشاشة، والرابع 20% من حجم الشاشة أما الثاني فسيكون حجمه بما تبقى من الشاشة.
<FRAMESET COLS="150,*,2*"> </FRAMESET>	يحدد ثلاثة إطارات الأول حجمه 150 بيكسل.... أما المساحة المتبقية فتقسم على أساس أن الإطار الثالث حجمه هو ضعفي (2*) حجم الإطار الثاني (*)

أما الخاصية الثانية فهي ROWS وأعتقد أنك استنتجت طبيعة عملها. نعم هي تحدد عدد وحجم الإطارات الأفقية (الصفوف) داخل الصفحة. وذلك بنفس الأسلوب المتبع مع الأعمدة، أي إما باستخدام الطريقة النسبية أو المطلقة. وسأقوم بسرد بعض الأمثلة لتوضيحها (وأذكرك ثانية أن هذه الأمثلة غير مكتملة):

<FRAMESET ROWS="50%,50%"> </FRAMESET>	يحدد إطارين أفقيين ارتفاع كل منهما 50% من ارتفاع الشاشة
<FRAMESET ROWS="20%,50%,30%"> </FRAMESET>	يحدد ثلاثة إطارات أفقية ارتفاعاتها 20% و 50% و 30% على التوالي من ارتفاع الشاشة
<FRAMESET ROWS="50,120,*"> </FRAMESET>	يحدد ثلاثة إطارات أفقية الأول ارتفاعه 50 بيكسل، والثاني 120 بيكسل، والثالث سيكون بالإرتفاع المتبقي من الشاشة
<FRAMESET ROWS="50,*,15%,20%"> </FRAMESET>	يحدد أربعة إطارات أفقية ارتفاع الأول هو 50 بيكسل، والثالث 15% من ارتفاع الشاشة، والرابع 20% من ارتفاع الشاشة أما الثاني فسيكون ارتفاعه بما تبقى من ارتفاع الشاشة.
<FRAMESET COLS="*,2*"> </FRAMESET>	يحدد إطارين الثاني ارتفاعه ضعفي ارتفاع الأول

لم ننته بعد من ذكر كل الخصائص المتعلقة بالوسوم <FRAMESET> فلا زال هناك الكثير. ولكن من الضروري أن نقوم الآن بالإنتقال إلى وسم آخر للإطارات لأنه مرتبط إرتباطاً وثيقاً بالوسوم السابقة وخصائصها المذكورة أعلاه، وهي <FRAME> فما هو عمل هذا الوسم؟

حسنا، كل ما قمنا به حتى الآن هو تعريف مجموعة من الإطارات وخصائصها (فقط تعريف الإطارات) لكن لم نحدد ماهية هذه الإطارات ولا محتوياتها ولا مصادرها. تماما كما نقوم بتعريف صفحات الويب الإعتيادية وخصائصها في الوسم <BODY> دون أن يعني ذلك تحديد محتويات هذه الصفحات. فإذا أردنا فيما بعد إدراج صورة مثلاً نستخدم الوسم الخاص بذلك وهو <IMG SRC="imagname.ext">

وفي حالة الإطارات فإننا نستخدم الوسم <FRAME> وهو وسم مفرد أي ليس له وسم نهاية تماماً مثل <IMG>. وفيه نقوم بتحديد مصدر وخصائص كل ملف نريد إظهاره داخل أحد الإطارات. ويتم استخدام هذا الوسم مرات بنفس

عدد الإطارات المذكورة داخل <FRAMESET>. وسوف أقوم مباشرة باستخدام الخاصية SRC لتحديد مصدر الملف.

دعنا نقوم الآن بإتمام الشيفرة لبعض الأمثلة المذكورة أعلاه. ونبدأ بالمثال الأول:

```
<FRAMESET COLS="50%,50%">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
</FRAMESET>
```

الآن ... والآن فقط أصبح لديك صفحة إطارات محترمة.

مثال آخر:

```
<FRAMESET COLS="200,400,*">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
  <FRAME SRC="frame3.html">
</FRAMESET>
```

مثال ثالث:

```
<FRAMESET ROWS="50,*,15%,20%">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
  <FRAME SRC="frame3.html">
  <FRAME SRC="frame4.html">
</FRAMESET>
```

ورابع:

```
<FRAMESET COLS="*,2*">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
</FRAMESET>
```

وبالإضافة إلى ما ذكر، نستطيع إدراج صورة مباشرةً داخل الإطار وباستخدام <FRAME SRC> تماماً كما ندرجها باستخدام <SRC IMG> وإليك هذا المثال:

```
<FRAMESET COLS="50%,50%">
  <FRAME SRC="frame1.html">
  <FRAME SRC="thedome.jpg">
</FRAMESET>
```

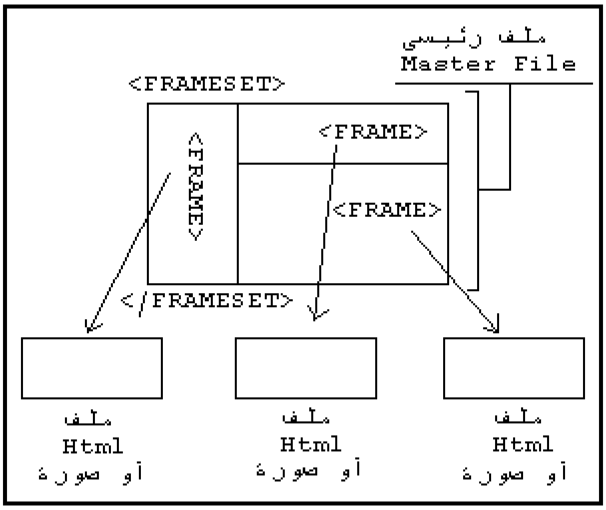
والآن لماذا لا نقم معاً بمراجعة الأفكار الأساسية السابقة الذكر وتلخيصها؟ وهذه هي:-

- لإدراج صفحة إطارات نحتاج إلى ملف رئيسي يعتبر بمثابة الوعاء الذي سيضم هذه

الإطارات.

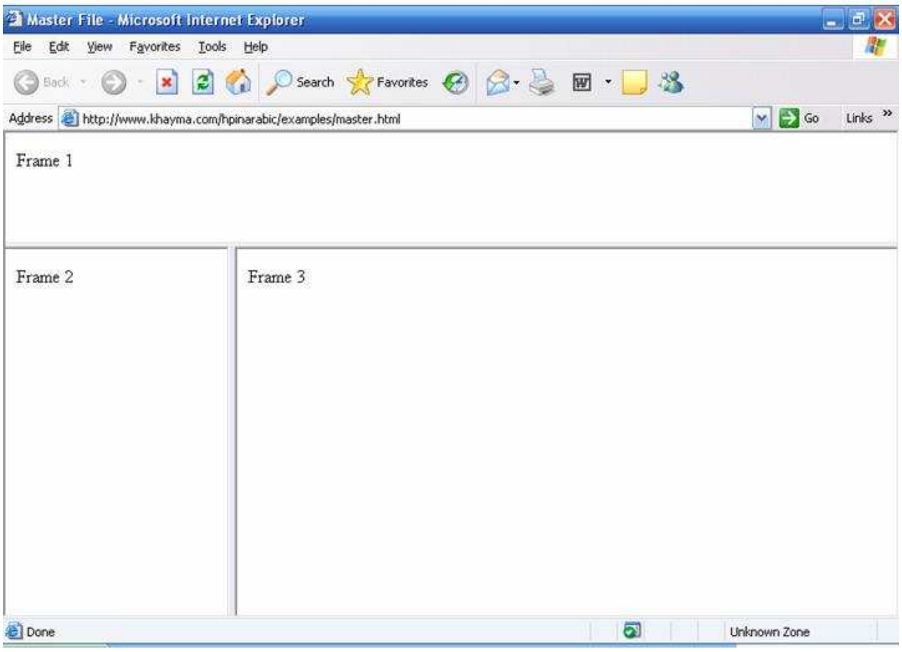
- الملف الرئيسي هو ملف HTML إعتيادي غير أننا نكتب الوسوم <FRAMESET> ...</FRAMESET> بدلاً من <BODY> ...</BODY>. وبالتالي فهو يحتوي على الخصائص التي نريدها للإطارات وتعريفاتها.
- نستخدم الخصائص COLS, ROWS لتحديد عدد الإطارات (صفوفاً كانت أو أعمدة) وأحجامها.
- الملفات الفرعية التي تظهر ضمن الإطارات هي ملفات عادية كالتي قمنا بإنشائها في الدروس السابقة أو صوراً. ولا تحتوي على أي تنسيق أو وسوم خاصة.
- نستخدم الوسم <FRAME> داخل الملف الرئيسي لمناداة الملفات الفرعية داخل الإطارات، وذلك مع الخاصية SRC. بالإضافة إلى استخدامه لتحديد باقي الخصائص .

كما نستطيع تمثيل هيكلية الإطارات من خلال الشكل التالي:

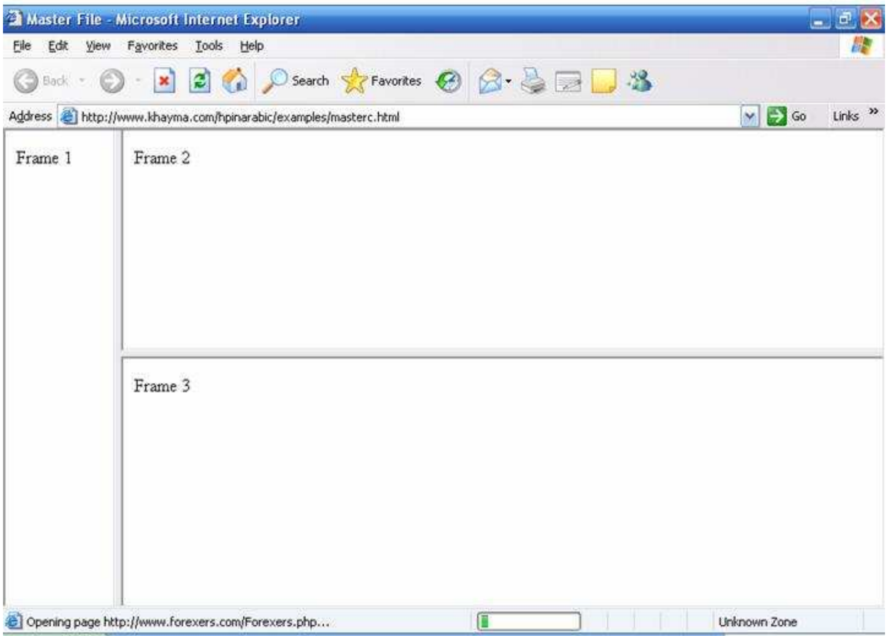


هل تأكدت من فهمك لهذه النقاط؟ لنتابع إذن ...
حتى الآن قمنا بتقسيم الصفحة إما لإطارات أفقية أو لإطارات عمودية. لكن نحتاج لمعرفة كيفية إدراج كلاهما في الصفحة. كما في الأمثلة التالية:

- صفحة مكونة من صفين، الثاني منهما مقسم بدوره إلى عمودين



- صفحة مكونة من عمودين، الثاني منهما مقسم بدوره إلى صفين



لنبدأ بالمثال الأول:
بما أن الصفحة تحتوي على صفين نقوم بتعريفهما أولاً حسب الإرتفاعات المرغوب بها:

```
<FRAMESET ROWS="100,*">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
</FRAMESET>
```

لكن الصف الثاني مقسم إلى عمودين وهنا يعتبر بمفهوم لغة HTML وكأنه صفحة إطارات جديدة لذلك لا نحتاج لتعريفه كصف وبدلاً من ذلك نعاود استخدام تعريف الصفحات! أي <FRAMESET> مرة أخرى.

```
<FRAMESET ROWS="100,*">
   <FRAME SRC="frame1.html">

   <FRAMESET>
   </FRAMESET>

</FRAMESET>
```

وبما أن الصف الثاني (أو لنقل الإطار الثاني) مقسم إلى عمودين، إذن بقي علينا إضافة تعريف لهذه الأعمدة. وبذلك تكون الشيفرة النهائية كالتالي:

```
<FRAMESET ROWS="100,*">
   <FRAME SRC="frame1.html">

   <FRAMESET COLS="200,*">
      <FRAME SRC="frame2.html">
      <FRAME SRC="frame3.html">
   </FRAMESET>

</FRAMESET>
```

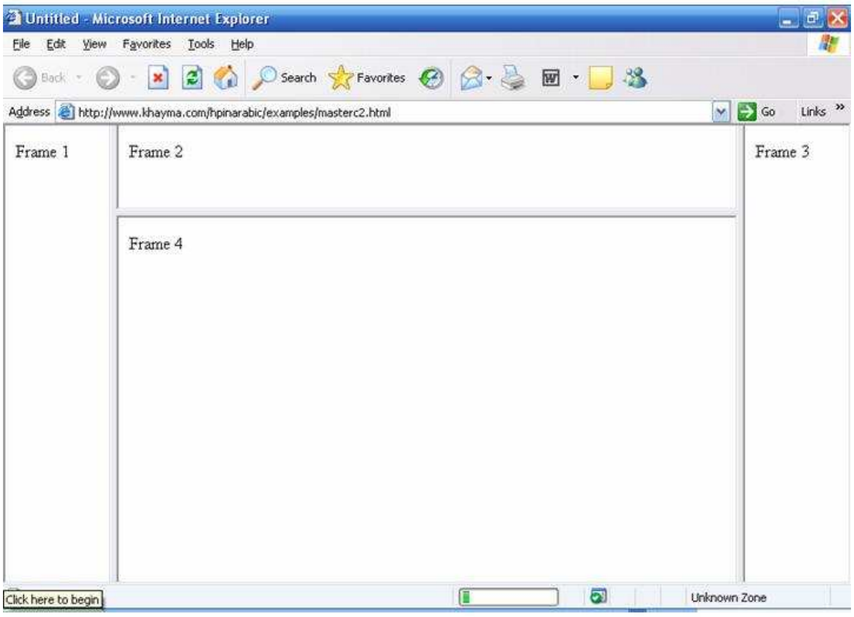
الشيفرة الخاصة بالمثال الثانيهمة فهذه هي

```
<FRAMESET COLS="100,*">
   <FRAME SRC="frame1.html">

   <FRAMESET ROWS="200,*">
      <FRAME SRC="frame2.html">
      <FRAME SRC="frame3.html">
   </FRAMESET>

</FRAMESET>
```

لنقم الآن بإدراج مثال آخر وتحليله:



يوجد لدينا ثلاثة أعمدة، أليس كذلك؟ إذن لنقم بتعريف صفحة إطارات ذات ثلاثة أعمدة (طبعاً لا يوجد أهمية للأحجام المذكورة، فأنا اخترتها حسب رغبتي وتستطيع أنت اختيار الأحجام التي تريدها).

```
<FRAMESET COLS="100,*,100">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
  <FRAME SRC="frame3.html">
</FRAMESET>
```

العمود الأوسط من هذه الصفحة مقسم إلى صفين، إذن نستبدل تعريفه بتعريف آخر لصفحة إطارات مكونة من صفين (وهذا هو التعريف بصورة مستقلة)

```
<FRAMESET ROWS="80,*">
  <FRAME SRC="frame2.html">
  <FRAME SRC="frame4.html">
</FRAMESET>
```

وبعد دمج الشيفرتين السابقتين معاً نحصل على هذه الشيفرة النهائية:

```
<FRAMESET COLS="100,*,100">
  <FRAME SRC="frame1.html">

  <FRAMESET ROWS="80,*">
    <FRAME SRC="frame2.html">
    <FRAME SRC="frame4.html">
  </FRAMESET>

  <FRAME SRC="frame3.html">
</FRAMESET>
```

وصلنا الآن إلى نهاية وقد قمنا في هذا الدرس بمناقشة أساسيات إدراج الإطارات، ولكن بقي هناك الكثير ليقال في هذا المجال. وهو ما سنكمله في الدرسين القادمين

الدرس العاشر
الإطارات (2)

لا زلنا نناقش موضوع الإطارات . لقد تعرفت في الدرس السابق على الأساسيات في هذا الموضوع، وتعلمت كيفية إنشاء صفحة ويب باستخدام مبدأ الإطارات. ونتابع معاً التعرف على باقي الخصائص والتحديدات المتعلقة بها.

بداية، أجد من الضروري أن أذكرك بالوسمين الأساسيين للإطارات واللذين ندرجهما في الملف الأساسي، وهما <FRAMESET> والذي يوضع بدلاً من <BODY> ويشكّل الوسم الأساسي لتحديد مجموعة الإطارات في الصفحة وخصائص هذه المجموعة ككل. و <FRAME> الذي يوضع داخل نطاق الوسم السابق ويعرّف الملف المصدر لكل إطار ويحدد خصائصه. ومن المهم جداً أن نميز بين الخصائص المتعلقة بكل وسم وأن لا نخلط بينهما.

وعند الحديث عن الإطارات سوف نواجه تلك المشكلة العتيدة التي تؤرق دائماً مصممي صفحات الويب، ألا وهي مسألة توافق الوسوم والخصائص مع المتصفحات المختلفة (والتي قمت بتوضيحها في الدرس السادس عشر).

فمثلاً لدينا أربع خصائص للوسم <FRAMESET> ، لكن واحدة منها فقط تعمل مع كلا المتصفحين الرئيسيين Netscape, MS Explorer. وكما إعتدنا سوف لن أركز على هذه الخصائص ، لكن على الأقل سأكتفي بذكرها وذكر عملها وأترك لك حرية تجربتها إن أردت.

أولى هذه الخصائص هي FRAMEBORDER وهي تقوم بتحديد ظهور أو عدم ظهور الحدود حول الإطارات وتأخذ القيم 1 للظهور، و0 لعدم الظهور. وهذا مثال عليها:

```
<FRAMESET ROWS="50,*,15%,20%" FRAMEBORDER="0">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
  <FRAME SRC="frame3.html">
  <FRAME SRC="frame4.html">
</FRAMESET>
```

أما الخصائص الثلاث الأخرى فهي:

- BORDER: تحدد سمك الحدود الظاهرة حول الإطارات وهي تأخذ قيماً بالبيكسل. (BORDER="n") وتعمل فقط مع Netscape
- BORDERCOLOR: لإضافة لون للحدود (BORDERCOLOR="rrggbb") وتعمل مع Netscape أيضاً.
- FRAMESPACING: لتحديد مسافات فارغة إضافية حول الإطارات وتأخذ قيماً بالبيكسل (FRAMESPACING="n") وهي تعمل مع MS Explorer.

أما الخصائص المستخدمة مع الوسم <FRAME> فهي كالتالي:

- MARGINHEIGHT: تحدد مقدار المسافة الفارغة المتروكة للهوامش العلوية والسفلية للإطار (بالبيكسل).

```
MARGINHEIGHT="n"
```

- MARGINWIDTH: تحدد مقدار المسافة الفارغة المتروكة للهوامش اليمنى واليسرى للإطار (بالبيكسل).

```
MARGINWIDTH="n"
```

- SCROLLING: تحدد إمكانية ظهور أو عدم ظهور أشرطة التصفح الأفقية والعمودية على جوانب أو أسفل الإطار. وتأخذ القيم yes للظهور. no لعدم الظهور. و auto التي تحدد ظهور الأشرطة أو عدمه تلقائياً بحسب الحاجة إليها. تماماً كما يحدث في معظم تطبيقات Windows

```
SCROLLING="yes"
SCROLLING="no"
SCROLLING="auto"
```

- NORESIZE عند إضافة هذه الخاصية يتم منع عملية التحكم بحجم الإطار بالتصغير أو التكبير من خلال السحب والإفلات. وهي لا تأخذ أي قيم.

ولتوضيح مبدأ عمل هذه الخصائص إليك هذه الأمثلة، (سأقوم باستخدام الملف الرئيسي الذي أدرجت فيه الصورة في الدرس السابق، فهو أفضل مثال لتوضيحها). وقم بتفحصه والتدقيق في تفاصيله لكي تقارنها بما سينتج عن الأمثلة التالية، كذلك حاول القيام بتكبير الإطار أو تصغيره بوضع المؤشر على الحد ثم استخدام السحب والإفلات بالإتجاه المطلوب. وذلك لكي تستطيع تمييز عمل الخاصية NORESIZE

```
<FRAMESET COLS="50%,50%">
<FRAME SRC="thedome.jpg" MARGINHEIGHT="40">
<FRAME SRC="frame2.html">
</FRAMESET>
```

```
<FRAMESET COLS="50%,50%">
<FRAME SRC="thedome.jpg" MARGINHEIGHT="40" MARGINWIDTH="30">
<FRAME SRC="frame2.html">
</FRAMESET>
```

```
<FRAMESET COLS="50%,50%">
<FRAME SRC="thedome.jpg" MARGINHEIGHT="40" MARGINWIDTH="30" SCROLLING="yes">
<FRAME SRC="frame2.html">
</FRAMESET>
```

```
<FRAMESET COLS="50%,50%">
<FRAME SRC="thedome.jpg" MARGINHEIGHT="40" MARGINWIDTH="30" SCROLLING="yes" NORESIZE>
<FRAME SRC="frame2.html">
</FRAMESET>
```

كذلك هناك الخصائص FRAMEBORDER, FRAMESPACING, BORDER, BORDERCOLOR التي تستخدم مع هذا الوسم وبنفس التفاصيل التي ذكرت مع <FRAMESET>. لكنها بالطبع تحدد خصائص الإطار وحده وليس مجموعة الإطارات ككل في الصفحة. وهي تعمل على متصفحات معينة دون غيرها.

بقي لدينا الخاصية NAME والتي تعتبر أهم خاصية لهذا الوسم، فهي التي تحدد طريقة تنسيق العمل بين الإطارات والصفحات وأسلوب عرضها لذلك فقد فضلت أن أفرد لها موضوعاً خاصاً وبصورة مستقلة عن باقي الخصائص، وذلك في الدرس التالي إن شاء الله.

هناك وسم ثالث يتعلق بالإطارات، ويتم إدراجه داخل الملف الرئيسي وعادة في النهاية وهو:

<NOFRAMES> ... </NOFRAMES>

يستخدم هذا الوسم لتوفير بديل معين عن صفحة الإطارات في حالة قيام أحد الزوار بدخول الموقع مستخدماً متصفحاً لا يدعم الإطارات. (بالمناسبة فإن هناك متصفحات لا يمكن لها أن تعرض الإطارات مثل الإصدارات القديمة لـِ Netscape, MS Explorer) لكنك حتماً تستطيع مشاهدتها وذلك لأن الإصدارات المعربة من هذه المتصفحات هي إصدارات حديثة نسبياً وتدعم الإطارات.

فإذا أردت أن تمنح زوار موقعك الذين لا يستخدمون متصفحاً حديثاً فرصة مشاهدة موقعك، فكل ما عليك فعله هو إدراج هذا الوسم في نهاية الملف الرئيسي للإطارات والبدء بكتابة صفحتك كما لو كانت صفحة ويب عادية.

```
<HTML>
<HEAD>
<TITLE>Main File</TITLE>
</HEAD>
<FRAMESET ROWS="50,*,15%,20%" FRAMEBORDER="0">
  <FRAME SRC="frame1.html">
  <FRAME SRC="frame2.html">
  <FRAME SRC="frame3.html">
  <FRAME SRC="frame4.html">
</FRAMESET>

<NOFRAMES>
<BODY>

                    أكتب صفحتك
                     بالصورة
                   الإعتيادية هنا

</BODY>
</NOFRAMES>

</HTML>
```

أما إذا كنت مصراً على إطاراتك ولا تريد إنشاء نسخة أخرى للموقع بدونها، فلم لا تكتب ملاحظة بسيطة ضمن هذا الوسم تخبر فيها زائرك بأن الموقع يحتوي على هذه الإطارات وأنه يحتاج إلى متصفح مناسب (على الأقل لكي تخفف عنه الصدمة)!!

وماذا بعد...؟ لم يبق أي شيء ليذكر في هذا الدرس فقد قمنا بمناقشة جميع خصائص الإطارات عدا الخاصية NAME. ما رأيك لو قمنا بالتدرب على إنشاء نسخة من هذا الموقع باستخدام الإطارات في الدرس التالي؟

الدرس الحادي عشر
الإطارات (3)

كما ترى من العنوان لا زلنا نتابع في موضوع الإطارات، وفي القسم الثالث منه. وذلك لأهمية هذا الموضوع وتشابك خطوطه. وخصوصاً لأننا نحتاج إلى التعامل مع أكثر من وسم في نفس الوقت وداخل أكثر من ملف. مما يستدعي التركيز والتروي عندما نريد إنشاء صفحة إطارات.

لقد أنهينا في الدرسين الماضيين شرح كافة الوسوم والخصائص التي تتعلق بالإطارات، عدا واحدة هي NAME ووعدتك أن أقوم بشرحها بصورة وافية في درس مستقل. كما وعدتك أن نقوم بإجمال موضوع الإطارات والتدرب عليها من خلال مثال واقعي يتمثل بإنشاء صفحة إطارات لهذا الموقع وهاأنا ذا أفي بوعدي.

لقد قلت إن هذه الخاصية تعتبر من أهم خصائص الوسم <FRAME> لأنها تحدد طريقة تنسيق العمل بين الإطارات والصفحات وأسلوب عرضها. **لنعتبر هذا تعريفاً مبدئياً**. ولكي أوضح لك الهدف من هذه الخاصية، قم باستعراض المثال التالي http://www.khayma.com/hpinarabic/examples/mainfile.html الذي قمت فيه بتمثيل بعض الأجزاء من هذا الموقع (لنترك تلك الأمثلة المملة التي كنا نعمل عليها في الدرسين السابقين ولنبدأ بالعمل الجدي). وأنقر على أزرار الوصلات التشعبية الموجودة في الإطار.
ماذا وجدت؟!؟ إن الصفحات تظهر في نفس الإطار (أعرف أنك كنت تتوقع ظهورها في الإطار الآخر... فهل خيبت ظنّك؟). حسناً، لا زالت هذه الإطارات تحتاج إلى القليل من العمل لكي تعمل كما يجب. وقبل أن نكمل أود أن تتمعن في الشيفرات التالية، وهي في الواقع الشيفرات الخاصة بالملفات المستخدمة في هذا المثال. وأرجو أن تقوم بدراستها جيداً، على الأقل لكي تكون متأكداً من فهمك لكل الأفكار السابقة قبل أن تنتقل إلى النقاط التي ستلي.

شيفرة الملف الرئيسي الذي أسميته mainfile.html

```
<HTML>
<HEAD>
<TITLE>Main File</TITLE>
</HEAD>
<FRAMESET ROWS="60,*">
<FRAME SRC="header.html" NORESIZE SCROLLING="NO" MARGINHEIGHT="1" MARGINWIDTH="1">
<FRAMESET COLS="120,*">
<FRAME SRC="lftframe.html" NORESIZE SCROLLING="NO" MARGINHEIGHT="1" MARGINWIDTH="1">
<FRAME SRC="htmintro.html">
</FRAMESET>
</FRAMESET>
</HTML>
```

شيفرة الملف العلوي (الترويسة) الذي أسميته header.html

```
<HTML>
<HEAD>
<TITLE>Header File</TITLE>
</HEAD>
<BODY bgcolor="#BCD8EB">
<CENTER><IMG SRC="frambnr.jpg"></CENTER>
```

```
</BODY>
</HTML>
```

شيفرة الملف الذي يحتوي على الأزرار والوصلات التشعبية والذي أسميته lftframe.html

```
<HTML>
<HEAD>
<TITLE>Buttons</TITLE>
</HEAD>
<BODY BGCOLOR="#BCD8EB">
<A HREF="htutor01.html"><IMG SRC="lesson1.jpg"></A><BR>
<A HREF="htutor02.html"><IMG SRC="lesson2.jpg"></A><BR>
<A HREF="htutor03.html"><IMG SRC="lesson3.jpg"></A><BR>
<A HREF="htutor04.html"><IMG SRC="lesson4.jpg"></A><BR>
<A HREF="htutor05.html"><IMG SRC="lesson5.jpg"></A><BR>
</BODY>
</HTML>
```

وبالطبع بقي لدينا الإطار الأكبر وهو لا يحتوي على ملف خاص لأنه الإطار العام الذي نريد إظهار الملفات فيه. ولن يتم إجراء أي تعديل على تلك الملفات. وبصورة مبدئية لقد شاهدت أنه يحتوى على صفحة المقدمة والمسماه htmintro.html

---

وهنا يأتي دور الخاصية NAME والتي نقوم من خلالها بتعيين اسم ما للإطار –**أي الإطار الذي نرغب أن تظهر به الملفات عندما نقوم بالنقر على الوصلات التشعبية**– حيث سيتم فيما بعد استخدام هذا الإسم من أجل استهداف هذا الإطار من قبل الوصلات التشعبية في الإطارات الأخرى أو حتى في الصفحات الأخرى.
وبالمناسبة فقد حان الآن ذكر خاصية جديدة من خصائص الوسم <A> ... </A> (وهو وسم الوصلات التشعبية كما عرفت في الدرس السادس.) وهذه الخاصية هي TARGET. (انتظر قليلاً وستعرف لماذا)

دعنا نقوم بترتيب الأمور بصورة أكثر وضوحاً وتسلسلاً:
برأيك بأي من الملفات الثلاثة السابقة يجب أن نضع الخاصية NAME؟ سؤال سخيف! أليس كذلك؟ طالما أنها إحدى خصائص الوسم <FRAME> إذن من البديهي جداً أن ترد معه. وبما أن هذا الوسم يوضع فقط في الملف الرئيسي إذن هي توضع في الملف الرئيسي... هذه أصبحت واضحة! لكن مع أي وسم من وسوم <FRAME> الموجودة في الملف الرئيسي؟؟ بالطبع ليس مع الوسم الخاص بتعريف ملف الترويسة فهذا لا نريد استهدافه، وكذلك الأمر بالنسبة لتعريف ملف الوصلات التشعبية الذي لا نريد استهدافه أيضاً. إذن لم يبق لدينا إلا الوسم الخاص بتعريف الإطار العام الذي ستظهر به الملفات. وبافتراض أننا نريد تسمية هذا الإطار بالاسم mainwindow. وهو اسم اختلقته أنا حسب ما أريد وأستطيع إعطاءه أي اسم آخر أريده طالما أنه يبدأ بحرف أبجدي أو رقم وليس برمز خاص مثل /؟#$%^ (عدا الرمز _ الذي يستخدم في حالات خاصة سأذكرها لك لاحقاً في هذا الدرس).

إذن تصبح شيفرة الملف الأساسي هي:

```
<HTML>
<HEAD>
<TITLE>Main File</TITLE>
</HEAD>
```

```
<FRAMESET ROWS="60,*">
<FRAME SRC="header.html" NORESIZE SCROLLING="NO" MARGINHEIGHT="1" MARGINWIDTH="1">
<FRAMESET COLS="120,*">
<FRAME SRC="lftframe.html" NORESIZE SCROLLING="NO" MARGINHEIGHT="1" MARGINWIDTH="1">
<FRAME SRC="htmintro.html" NAME="mainwindow">
</FRAMESET>
</FRAMESET>
</HTML>
```

وبهذا نكون قد إنتهينا من إعداد الملف الرئيسي ليكون ملف إطارات فعال مائة بالمائة. ونستطيع الآن استهداف ذلك الإطار من أي مكان باستخدام الإسم الذي عرفناه به.

المرحلة الثانية هي تعريف الوصلات التشعبية بطريقة تجعل الملفات المتعلقة بهذه الوصلات تظهر في الإطار المحدد. وهنا يأتي دور الخاصية TARGET سالفة الذكر لكي تقوم على الرحب والسعة بأداء هذه الوظيفة. وسوف نعمل الآن على الملف المسمى lftframe.html. أليس هو الملف الذي يحتوي على الوصلات التشعبية؟! ليصبح بالشكل التالي:

```
<HTML>
<HEAD>
<TITLE>Buttons</TITLE>
</HEAD>
<BODY BGCOLOR="#BCD8EB">
<A HREF="htutor01.html" TARGET="mainwindow"><IMG SRC="lesson1.jpg"></A><BR>
<A HREF="htutor02.html" TARGET="mainwindow"><IMG SRC="lesson2.jpg"></A><BR>
<A HREF="htutor03.html" TARGET="mainwindow"><IMG SRC="lesson3.jpg"></A><BR>
<A HREF="htutor04.html" TARGET="mainwindow"><IMG SRC="lesson4.jpg"></A><BR>
<A HREF="htutor05.html" TARGET="mainwindow"><IMG SRC="lesson5.jpg"></A><BR>
</BODY>
</HTML>
```

لقد نجحنا... أليس كذلك

الحقيقة أننا لم ننجح بصورة مطلقة، بل إلى حد ما... وأعتذر لك لأني نغصت عليك هذا النجاح. لكن طالما أننا نريد الوصول إلى أفضل مستوى وأعلى أداء لموقعنا فيجب علينا دائماً تطبيق القواعد التي تضمن ذلك. ولكي أوضح لك السبب الذي يجعلني أتكلم كالفلاسفة سوف أطلب منك العودة إلى الصفحة الرئيسية للإطارات بعد أن قمت بإضافة وصلات تشعبية إضافية لها، واحدة خاصة بالإنتقال إلى صفحة الإطارات نفسها، أي إعادة تحميل الصفحة على الشاشة. والثانية للإنتقال من طريقة العرض بالإطارات إلى العرض بدون إطارات وقم بتجربة النقر على هذه الوصلات الجديدة.

مشكلة جديدة، أليس كذلك؟ ففي الحالة الأولى تم تحميل الصفحة داخل الإطار نفسه، وبذلك أصبحت الصفحات متداخلة في ما بينها، وكذلك الأمر في الحالة الثانية. أي أننا بإختصار لا نستطيع إبقاء الوسم بالشكل:

```
<A HREF="filename.html" TARGET="mainwindow"> ... </A>
```

كما لا نستطيع كتابته بالشكل التالي:

```
<A HREF="filename.html"> ... </A>
```

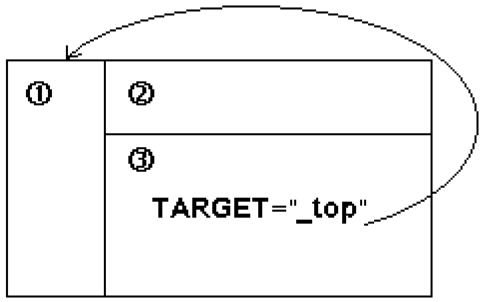
وطبعاً أنت تعرف السبب في كلتا الحالتين.

لقد وقعنا في مصيدة الإطارات. إذن فما الحل؟ الحل توفره لغة HTML نفسها من خلال وضع قيمة معينة للخاصية TARGET وهي "top_" لتخبر المتصفح أن يقوم بتحميل الصفحة في **المستوى الأعلى** من الشاشة. أي ان يقوم بإلغاء أي إطارات أو صفحات عادية موجودة أصلاً وأن يحمل الصفحة الجديدة مكانها. وهذه القيمة معرفة بشكل مسبق في لغة HTML وليست من عندي كما في الإسم mainwindow.

والحقيقة أنه يوجد ثلاث قيم أخرى بالإضافة إلى top_ معرفة بشكل مسبق في لغة HTML لتحدد موقع ظهور الصفحة المعنية. ويجب أن تتذكر دائماً أنها تكتب بالأحرف الصغيرة lowercase. فإذا كتبت بالأحرف الكبيرة فلن تحصل على النتيجة التي تريدها لأنها ستعتبر في هذه الحالة وكأنها أسماء عادية. وهذه هي القيم الأربعة:

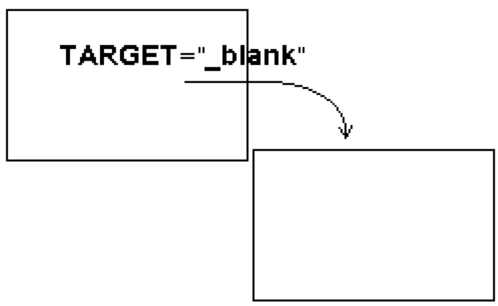
_top

تحمل الصفحة في أعلى مستوى للشاشة أي تلغي الإطارات الموجودة أصلاً



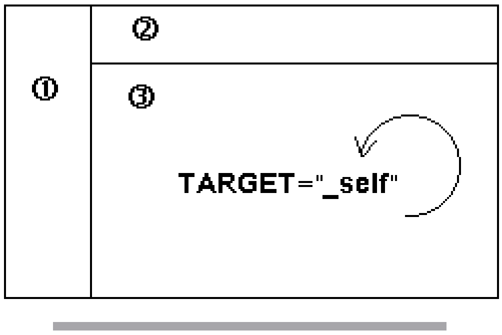
_blank

تقوم بفتح شاشة جديدة وفارغة للمتصفح وتعرض الصفحة المحملة فيه



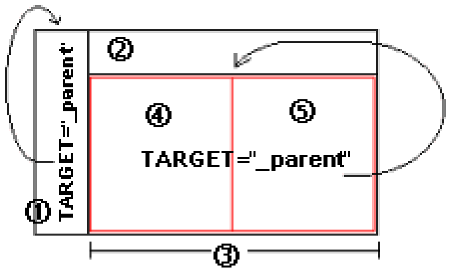
_self

(الحالة الإفتراضية) تقوم بتحميل الصفحة في نفس الإطار الذي توجد به الوصلة التشعبية التي تم النقر عليها، (وبالطبع هذا ما يحدث دائما دون وجود هذه القيمة، أليس كذلك؟)



_parent

تقوم بعرض الإطار المحمل مكان الإطار الأب،
وتأمل الشكل التالي الذي يمثل صفحة إطارات مركبة.



أنت تعرف أننا نحتاج إلى ثلاثة أزواج من الوسوم <FRAMESET> ... </FRAMESET> لإخراج مثل هذه الصفحة.

- الأول لتحديد الصفحة ككل
- الثاني لتحديد الإطارين 2 و 3 المتفرعين من الصفحة الرئيسية
- الثالث لتحديد الإطارين 4 و 5 المتفرعين من الإطار 3

وبالتالي نقول أن الصفحة الرئيسية هي الأب (أو الأم إذا أردت) للإطارين 2 و 3 وأن الإطار 3 هو الأب للإطارين 4 و 5. وعلى سبيل المثال إذا احتوى الإطار 5 أو 4 على وصلة تشعبية هدفها القيمة _parent لتم تحميلها مكان الإطار 3... وهكذا الأمر بالنسبة للإطارين 1 و 2 بمقابل الصفحة الرئيسية.
وللتمييز بين هذه القيمة والقيمة _top فإن القيمة _top تقوم بالتحميل في المستوى الأول (الأعلى) دائماً وبغض النظر عن موقع الوصلة التشعبية.

ونعود الآن إلى صفحتنا. فلو أضفنا السطرين التاليين إلى ملف الوصلات التشعبية:

```
<A HREF="mainfile.html" TARGET="_top"><IMG SRC="mainpage.jpg"></A><BR>
<A HREF="htmintro.html" TARGET="_top"><IMG SRC="noframes.jpg"></A><BR>
```

لحصلنا على النتيجة المرجوة. جرّب ذلك وبذلك تستطيع أن تقول وبكل ثقة

نجحنا بجدارة

الدرس الثاني عشر
وسوم ... من هنا وهناك!!!

كما ترى من العنوان فقد اخترت أن يكون هذا الدرس خارجاً قليلاً عن نطاق الدروس السابقة من حيث المحتوى. فلن تجد هنا وسوماً محددة تستطيع حصرها تحت موضوع معين، بل وسوماً عامة وخصائص إضافية لوسوم ذكرت سابقاً. وإن كان الهدف من معظمها ترتيب الصفحة والتحكم بشكل محتوياتها. ولا أخفي عليك أني قصدت تأجيل بعضها حتى هذا الدرس مع أنه كان من الممكن إدراجها ضمن دروس سابقة، وخاصة الدرس الرابع (الفقرات). لكن حجّتي في عدم إدراجها في حينه أن هذه الوسوم لن يتم استخدامها ولن تُفهم طريقة عملها إلاّ عند استخدام وسوم أخرى تمّ شرحها في وقت لاحق بعد الفقرات مثل الصور والجداول. أما البعض الآخر ففضلت عدم حصرها ضمن أي درس على اعتبار أنها وسوم عامة لا تختص بأي موضوع. على أية حال يكفينا هذه المقدمة وأترك لك حرية تصنيفها كما يحلو لك.

أترى هذا الخط الذي فصلت به هذه الفقرة عن الفقرة السابقة؟ إنه يسمى بلغة HTML بالمسطرة الأفقية Horizontal Rule وتستطيع إدراجه لتقسيم صفحتك بكتابة الوسم <HR> فقط لا غير. أكتب:

```
<HR>
```

ليظهر لديك هذا الخط:

لكن هذا ليس كل شيء. لأنك تستطيع تحديد سُمك هذا الخط إذا أضفت له الخاصية SIZE وأتبعتها برقم يمثل هذا السُمك مثلاً:

```
<HR SIZE="5">
```

```
<HR SIZE="1">
```

```
<HR SIZE="10">
```

كذلك يمكنك تحديد عرض الخط باستخدام الخاصية WIDTH والتي من الممكن أن تأخذ قيمة مطلقة أو نسبية

```
<HR WIDTH="80%">
```

```
<HR WIDTH="400">
```

```
<HR SIZE="5" WIDTH="60%">
```

ومن الخصائص الأخرى لهذا الخط خاصية المحاذاه ALIGN والتي تأخذ القيم center, left, right

```
<HR WIDTH="80%" ALIGN="center">
```

```
<HR WIDTH="400" ALIGN="left">
```

```
<HR SIZE="5" WIDTH="60%" ALIGN="right">
```

وكما تلاحظ فإن هذا الخط يكتسب مظهراً غائراً ثلاثي الأبعاد وإذا أردت خطاً عادياً غير غائر فقم بإضافة الخاصية NOSHADE

```
<HR SIZE="5" WIDTH="60%" ALIGN="center" NOSHADE>
```

أما إذا كان لون هذا الخط لا يعجبك، فما من مشكلة إذ أنك تستطيع اختيار اللون الذي يعجبك من خلال الخاصية COLOR (تعمل فقط مع Explorer MS)

```
<HR SIZE="5" WIDTH="60%" ALIGN="center" COLOR="#FF0000" NOSHADE>
```

الوسم التالي في هذه المجموعة هو وسم الملاحظات <!-- ... --> ونستخدمه عند الحاجة لكتابة بعض الملاحظات الخاصة أو العبارات التوضيحية ضمن الملف والتي يقصد أن لا تظهر عند استعراض هذا الملف في المتصفح.

```
This is line one<BR>
<!-- This is line two --><BR>
and, this is line three<BR>
```

وهذه هي النتيجة

This is line one

and, this is line three

من المؤكد أنك تعرف الوسم
 والذي يقوم بالتحكم في نهايات الأسطر (أي أنه ينهي السطر الحالي بحيث يظهر النص الذي يليه في سطر جديد). فهل تعلم أنه يوجد خاصية لهذا الوسم وهي CLEAR؟

لكي تتوضح لك طبيعة عمل هذه الخاصية، قم بمراجعة ما ذكرناه في الدرس الخامس عن الصور والرسومات، واستخدام الخاصية ALIGN التي تحدد موقع هذه الصور على الصفحة. حسناً، لقد إتفقنا في حينه على أن القيمة right توجه الصورة إلى يمين الصفحة وأن النص الذي يليها يظهر ملتفاً بعدة أسطر على الجهة اليسرى. وكذلك الأمر

(بصورة معكوسة) عند استخدام القيمة left. وحتى لو استخدمنا القيمة bottom أو لم نقم بإضافة الخاصية ALIGN أصلاً، فسوف نجد أن النص الذي يليها يظهر بمحاذاة الحافة السفلى للصورة.

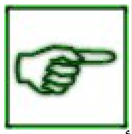
يتلخص عمل الخاصية CLEAR في منع النص من الإلتفاف على أي من جانبي الصورة.
وهي تأخذ القيم right التي تمنع ظهور النص إلاّ عند بداية الهامش الأيمن الفارغ بعد الصورة (أي بمنتهى البساطة تمنع ظهور النص والتفافه على الجهة اليسرى للصورة... أليست هذه العبارة أسهل للفهم؟!!) وبالتالي فإن هذه القيمة تستخدم عندما تكون قيمة ALIGN للصورة هي right.

وبنفس المبدأ لكن بصورة معكوسة نستخدم القيمة left والتي في نتيجتها تمنع إلتفاف النص على الجهة اليمنى من الصورة. وذلك عندما تكون محاذاة الصورة هي left

منطق معكوس... أليس كذلك؟ على أية حال إذا كنت تكرة هذه التعقيدات وتحب الطرق المختصرة مثلي، فاستخدم القيمة all التي تمنع الإلتفاف من جميع الجوانب.

ما رأيك أن نقتبس بعض الأمثلة من الدرس الخامس لنشاهد كيف تعمل مع إضافة هذه الخاصية؟ وسوف أدرج هذه الأمثلة كما هي نصاً وحرفاً أما الإضافات فهي المميزة باللون الأحمر

RIGHT



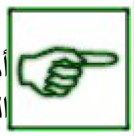
أما هذه القيمة فتؤدي إلى محاذاة الصورة إلى أقصى اليمين. مع التفاف النص الذي يليها على الجهة اليسرى ولعدة أسطر حسب ارتفاع الصورة.

<IMG SRC="image.jpg" ALIGN="RIGHT">
<BR CLEAR="right">

فإذا أضفنا الوسم
 مع الخاصية CLEAR="right" لوجدنا أنها منعت النص من الإلتفاف

والآن لنجرب استخدام القيمة left مع هذا المثال نفسه

RIGHT



أما هذه القيمة فتؤدي إلى محاذاة الصورة إلى أقصى اليمين. مع التفاف النص الذي يليها على الجهة اليسرى ولعدة أسطر حسب ارتفاع الصورة.

<IMG SRC="image.jpg" ALIGN="RIGHT">
<BR CLEAR="left">

نلاحظ أن لا فائدة من استخدام الوسم
 مع الخاصية CLEAR="left" فكل ما فعلته هو إضافة سطر فارغ أعلى النص

حسناً لنستخدم القيمة left في مكانها الصحيح، أي مع المحاذاة left

LEFT



هذه القيمة تؤدي إلى محاذاة الصورة إلى أقصى اليسار. مع التفاف النص الذي يليها على الجهة اليمنى ولعدة أسطر حسب ارتفاع الصورة.

```
<IMG SRC="image.jpg" ALIGN="LEFT">
<BR CLEAR="left">
```

الآن تعمل هذه القيمة كما يجب (لا شيء أفضل من وضع الوسم المناسب في المكان المناسب!)

و أترك لك المجال لكي تجرب القيمة all بنفسك

من القواعد الإفتراضية للمتصفحات أن الأسطر في كل فقرة تلتف وتنقسم بصورة تلقائية حسب إستبانة الشاشة وعرض نافذة المتصفح. (هذه نعرفها من الدرس الأول). لكن لنقل أننا نريد من أحد الأسطر أن لا ينقسم مهما كان مقدار الإستبانة و عرض النافذة.
حسناً، كل ما علينا فعله هو وضع هذا السطر ضمن الوسوم

<NOBR> ... </NOBR>

و هي إختصار لـِ NO BReak أي (لا إنقسام).

الدرس الثالث عشر
الخرائط الصورية

مع بداية هذا الدرس تكون قد قطعت شوطاً طويلاً مع هذه اللغة، وأصبحت قادراً على إنشاء صفحات الويب بصورة فعالة. وسيخصص هذا الدرس لمناقشة واحداً من المواضيع المتقدمة نوعاً ما والممتعة (برأيي المتواضع) في لغة HTML ألا وهو موضوع الخرائط الصورية.

أنت تعرف بلا شك من خلال دراستك للوصلات التشعبية، أننا نستطيع إدراج أي صورة نريدها لتمثل وصلة تشعبية ما. لكن ما رأيك بصورة واحدة تحتوي على العديد من هذه الوصلات التي تقوم بإيصالنا إلى مواقع مختلفة!!؟ هذا هو ببساطة مبدأ الخرائط الصورية. من المحتمل أن تكون لا تعرفها وأن يكون هذا الموضوع جديداً عليك، لكن من المؤكد أنك استخدمتها من قبل.

إذن، إضافةً إلى الوسوم الجديدة الخاصة بالخرائط والتي سنتعرف عليها من خلال هذا الموضوع سوف نعود أيضاً للتعامل مع وسوم الصور (بما أننا نتحدث عن خرائط صورية) كذلك سنتعامل مع الوصلات التشعبية (كون الخرائط الصورية هي تطبيق آخر للوصلات).. وستحتاج أيضاً إلى القليل من الهندسة (نعم .. الهندسة. ذلك العلم الذي يجبرك المعلم فيه على حمل البيكار لترسم به دائرة.. فتكون النتيجة شكلاً عجائبياً قد يكون أي شيء .. إلا أن يكون دائرة). وأخيراً نحتاج إلى أحد برامج تحرير الصور مثل Paint Shop Pro لمساعدتنا في معرفة بعض التفاصيل الخاصة بالصور.

هل أنت مستعد؟ لننطلق إذن.

هذه هي خريطتنا أو بالأحرى ما سوف تصبح بعد قليل خريطتنا .. مجرد صورة عادية سنقوم بإدراجها في الصفحة.



وهذه هي الشيفرة الخاصة بإدراجها

```
<IMG SRC="flowers.jpg" WIDTH="320" HEIGHT="310" BORDER="0">
```

والآن سوف نضيف لهذه الشيفرة خاصية جديدة من خصائص الوسم <IMG> هي "USEMAP="#map_name والتي تخبر المتصفح أن يستخدم هذه الصورة كخريطة وذلك حسب التحديدات الموجودة في تعريف هذه الخريطة (وسنتحدث عن هذه التحديدات لاحقاً إن شاء الله). أما القيمة map_name فهي تمثل الاسم الذي نريد أن نعيّنه للخريطة (أي اسم نريده). وكما ترى فإن هذا الاسم مسبوق بإشارة #.

ما رأيك أن نسمي الخريطة باسم ourmap. عندها سوف تصبح الشيفرة كما يلي:

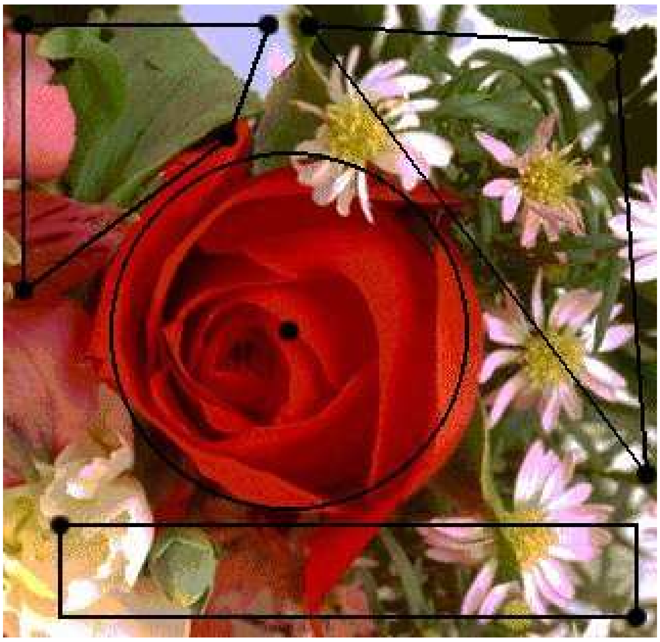
```
<IMG SRC="flowers.jpg" WIDTH="320" HEIGHT="310" BORDER="0" USEMAP="#ourmap">
```

وكانت هذه هي الخطوة الأولى في تعريف الخريطة. والخطوة الثانية هي تحديد التفاصيل.

لنقم بإخبار المتصفح أننا نريد خريطة تحمل الإسم ourmap

```
<MAP NAME="ourmap">
...
</MAP>
```

ولاحظ أننا لا نستخدم إشارة # مع التسمية هنا
بعد ذلك نحتاج أن نحدد المناطق الساخنة للوصلات التشعبية، ولنفترض أننا نريدها بالشكل التالي:



وهنا أريد أن أؤكد عليك وبصورة مشددة أن رسم هذه الأشكال الهندسية هو فقط بغرض توضيح مبدأ المناطق الساخنة لك، وتحديد شكل وموقع كل منطقة. ولا نحتاج لها أبداً عندما نريد إنشاء خريطة صورية كذلك فقد قمت باختيارها ورسمها بصورة عشوائية، بمعنى أنك تستطيع إختيار أي مناطق أخرى مغايرة كما تستطيع إختيار أي أشكال تريدها. وذلك بالطبع وفقاً لرغبتك كمصمم صفحات ويب.

والسؤال الآن هو، كيف نحدد هذه المناطق باستخدام HTML وكيف يمكن للمتصفح أن يميزها دون غيرها؟ والجواب هو استخدام الوسم <AREA> والذي يدرج ضمن الوسم السابق وبنفس عدد المناطق الساخنة التي نريدها.

```
<MAP NAME="ourmap">
```

```
<AREA>
<AREA>
<AREA>
<AREA>

</MAP>
```

ومع ذلك فهذا ليس كل شيء لأننا نحتاج أولاً لتحديد شكل كل منطقة ونستخدم لهذا الغرض الخاصية SHAPE ومن ثم نحتاج لتحديد موقع كل منطقة وهذا يتم بالخاصية COORDS وأخيراً بما أن كل منطقة تمثل وصلة تشعبية وتشير إلى عنوان موقع ما فمن البديهي أن نستخدم خاصية تحديد المواقع وهي HREF والتي مرت معنا في الدرس السادس

إلى التفاصيل...

تأخذ الخاصية SHAPE ثلاثة قيم لتحديد شكل المنطقة الساخنة وهي:

RECT
للأشكال المربعة والمستطيلة المنتظمة

CIRCLE
للدوائر

POLY
للأشكال العشوائية والمضلعة

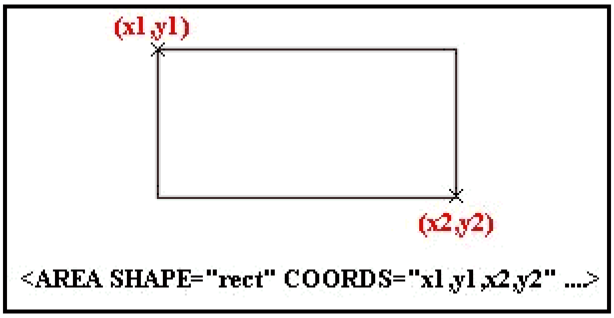
ولدينا دائرة ومستطيل وشكلين عشوائيين، إذن تصبح الشيفرة كما يلي (دون أهمية لترتيبها):

```
<MAP NAME="ourmap">

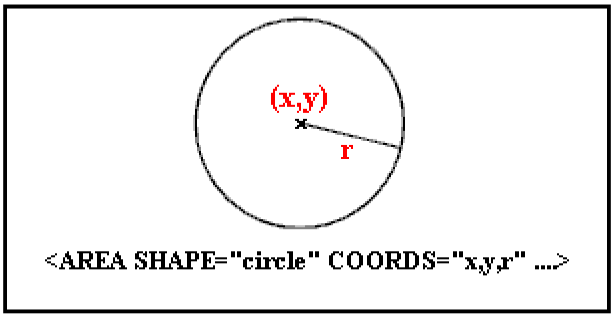
<AREA SHAPE="poly">
<AREA SHAPE="rect">
<AREA SHAPE="circle">
<AREA SHAPE="poly">

</MAP>
```

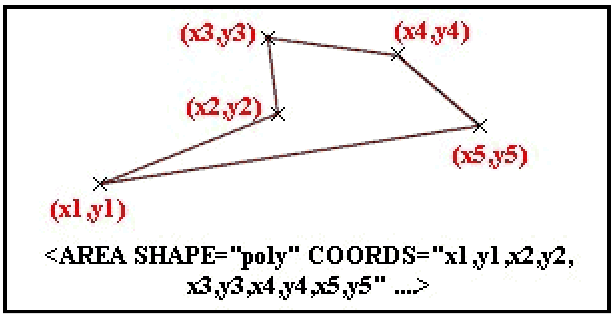
نأتي الآن إلى تحديد موقع كل شكل من الخريطة، والذي يتم باستخدام الخاصية COORDS. تأخذ هذه الخاصية قيماً رقمية تمثل إحداثيات الشكل الهندسي بالبيكسل على الخريطة (ألم أقل لك أنك ستحتاج إلى القليل من الهندسة) ومن البديهي أن تختلف دلالة كل قيمة بإختلاف الشكل المقصود. أنظر إلى الأشكال التوضيحية التالية:



في الشكل الرباعي المنتظم نحتاج لمعرفة إحداثيي الزاوية اليسرى العليا وإحداثيي الزاوية اليمنى السفلى. (أربعة أرقام)



أما في الدائرة فكل ما نحتاجه هو إحداثيي نقطة المركز لهذه الدائرة بالإضافة إلى معرفة نصف قطرها. (ثلاثة أرقام فقط)



وأخيراً في الأشكال المضلعة العشوائية نحتاج لمعرفة إحداثيي كل نقطة من النقاط التي تمثل زوايا هذا الشكل مهما كان عددها. (عدد الأرقام غير محدد ويختلف بإختلاف الشكل المقصود)

وقد تسأل الآن، كيف نستطيع إيجاد هذه الإحداثيات؟ ما من مشكلة! هنا يأتي دور برنامج الجرافيكس، فكل ما عليك هو تحميل الصور في هذا البرنامج ثم وضع المؤشر في المكان المطلوب وسوف تشاهد إحداثيات النقطة التي يقع عليها المؤشر على الشريط الموجود أسفل نافذة البرنامج. وبالإضافة لذلك تستطيع معرفة أبعاد الصورة ككل. وعلى أية حال كل ما عليك فعله هو تسجيل إحداثيات جميع النقاط المطلوبة من أجل استخدامها في تعريف الخريطة.



ولنعد الآن إلى خريطتنا:
بالنسبة للشكل المستطيل كانت إحداثيات النقطة العليا هي:(28,255) وإحداثيات النقطة السفلى هي (310,300).

أما الدائرة فكانت إحداثيات نقطة مركزها هي (140,160) ونصف قطرها 175 بيكسل.

أما الشكل المضلع الأيمن (المثلث) فأن إحداثيات زواياه هي (150,10), (300,20), (315,230) وتذكر هنا أن لا أهمية للترتيب في ذكر هذه النقاط. المهم فقط أن تكون الإحداثيات دقيقة إلى حد ما.

وأخيراً فإن إحداثيات النقاط في الشكل المضلع الأيسر (الرباعي) هي: (10,10), (130,10), (110,65), (10,140)

لنقم الآن بكتابة هذه البيانات ضمن الشيفرة السابقة والتي ستصبح كما يلي:

```
<MAP NAME="ourmap">

<AREA SHAPE="poly" COORDS="10,10,130,10,110,65,10,140">
<AREA SHAPE="rect" COORDS="28,255,310,300">
<AREA SHAPE="circle" COORDS="140,160,175">
<AREA SHAPE="poly" COORDS="150,10,300,20,315,230">

</MAP>
```

ماذا بقي الآن؟ بقي تعيين العناوين التي ستشير إليها الوصلات التشعبية والتي سنستخدم لها الخاصية HREF. إليك عناوين أربعة من المواقع العربية الممتازة

```
<MAP NAME="ourmap">

<AREA SHAPE="poly" COORDS="10,10,130,10,110,65,10,140"
HREF="http://www.sakhr.com">
<AREA SHAPE="rect" COORDS="28,255,310,300"
```

```
HREF="http://www.ayna.com">
<AREA SHAPE="circle" COORDS="140,160,175"
HREF="http://www.pcmag-arabic.com">
<AREA SHAPE="poly" COORDS="150,10,300,20,315,230"
HREF="http://www.asp.com.lb">

</MAP>
```

وأخيراً بعد تجميع الشيفرة الخاصة بإدراج الصورة مع هذه الشيفرة بالشكل التالي. (وبالمناسبة فقد قمت بالتجميع من أجل توضيح النتيجة النهائية لشيفرة الخريطة ككل. وفي الواقع أننا نستطيع وضع تعريف الخريطة في أي مكان نريده داخل الملف وليس شرطاً أن يكون قريباً من شيفرة إدراج الصورة)

```
<IMG SRC="flowers.jpg" WIDTH="320" HEIGHT="310" BORDER="0" USEMAP="#ourmap">

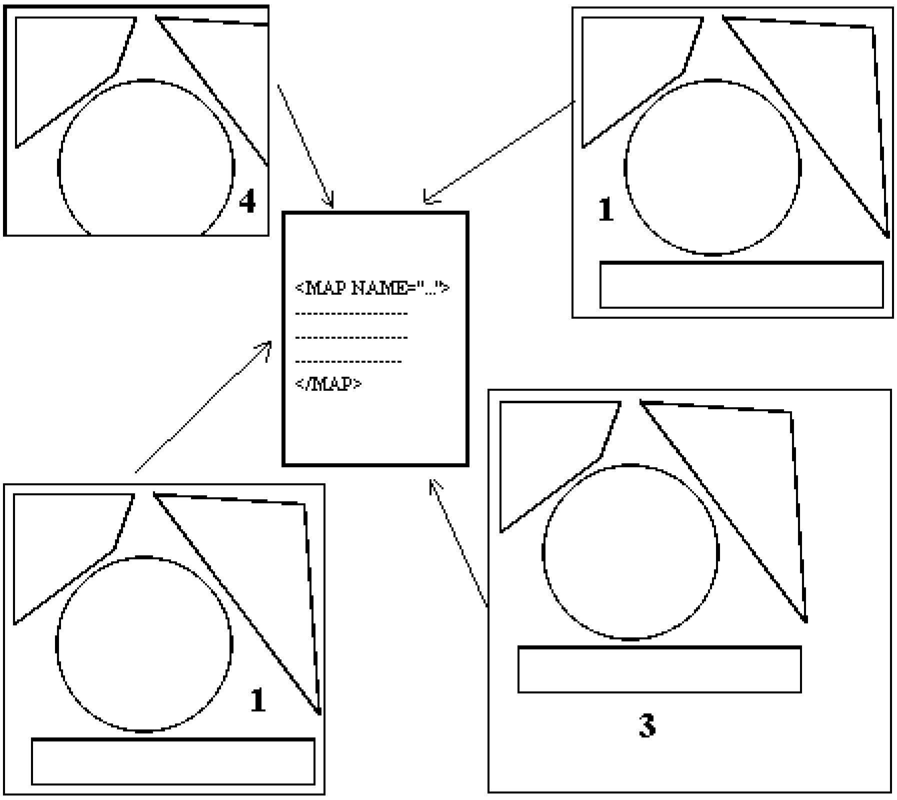
<MAP NAME="ourmap">

<AREA SHAPE="poly" COORDS="10,10,130,10,110,65,10,140"
HREF="http://www.sakhr.com">
<AREA SHAPE="rect" COORDS="28,255,310,300"
HREF="http://www.ayna.com">
<AREA SHAPE="circle" COORDS="140,160,175"
HREF="http://www.pcmag-arabic.com">
<AREA SHAPE="poly" COORDS="150,10,300,20,315,230"
HREF="http://www.asp.com.lb">

</MAP>
```

والآن أريد أن أثير مسألتين مهمتين في هذا السياق:
الأولى: ماذا لو أردنا إدراج هذه الخريطة أكثر من مرة في الصفحة، فهل نحتاج إلى تكرار كتابة الشيفرة <MAP> ... </MAP> مع كل مرة ندرج فيها الصورة؟ الجواب قطعاً لا.!!
فطالما أننا نكتب الخاصية USEMAP فإن المتصفح يقوم بالبحث عن شيفرة الخريطة التي حددنا اسمها مع هذه الخاصية، ومن ثم يطبقها على الصورة المعطاة مهما كان عدد مرات إدراجها.

الثانية: ماذا لو قررنا إلغاء هذه الصورة ووضع صورة أخرى مكانها. هل من مشكلة؟ أيضاً لا يوجد هنا أي مشكلة. لأن تعريف الخريطة بطبيعته غير مرتبط بصورة محددة بعينها. هو فقط يعرف مناطق محددة بالشكل والموقع. ولو دققت النظر في هذا التعريف فلن تجد ما يشير إلى أنها مرتبطة بصورة بعينها، أليس كذلك؟ بل إنك لو أدرجت صورة أكبر مما يجب أو حتى صورة أصغر مما يجب واستخدمت خاصية USEMAP معهما لتشير إلى الخريطة، فلن تواجه أي مشكلة في ذلك. (وأعني هنا مشكلة للمتصفح) لكن بالطبع فالصورة الكبيرة ستحتوي على مناطق دون وصلات تشعبية، والصورة الصغيرة سوف لن تحتوي على بعض الوصلات المعرفة في الخريطة



والخلاصة أن تعريف الخريطة هو تعريف مشاع يوضع مرة واحدة فقط وفي أي مكان داخل الصفحة، وبشرط أن يكون ضمن <BODY> ... </BODY> وبالمقابل نستطيع إدراج عشرات الصور مهما كانت طبيعتها داخل الصفحة لتكون خرائط صورية تستخدم نفس تعريف هذه الخريطة طالما أننا وضعنا الخاصية USEMAP="#map_name" مع هذه الصور.

ومسك الختام: تحفل الإنترنت بالكثير من البرامج التي تقوم بتصميم الخرائط الصورية بسرعة وسهولة متناهيتين.. فقط تقوم بتحميل الصورة المطلوبة، ثم ترسم الأشكال التي تريدها عليها كما لو كنت تستخدم برنامج الرسام، ومن ثم تكتب عناوين الوصلات التشعبية المطلوبة لكل من هذه الأشكال. وهكذا ببساطة يقوم البرنامج بإيجاد الإحداثيات الخاصة بكل شكل وإنشاء شيفرة الخريطة المطلوبة!! أي أنك خلال دقائق معدودة تستطيع إنشاء أعقد الخرائط الصورية.. لكن بما أنك تهوى الصعب وتتحدى المستحيل فلا اعتقد أنك ستفضل هذا النوع من البرامج. (بالمناسبة، لقد فضلت تأخير كتابة هذه الملاحظة حتى نهاية هذا الدرس، لأني لو وضعتها في البداية لما أكملت هذا الدرس أصلاً ولهرعت للإنترنت للبحث عن مثل هذه البرامج) وعلى أية إليك هذا العنوان لبرنامج رائع يدعى Live Image
http://www.mediatec.com/a/index.php أتمنى لك خرائطاً سعيدة دائماً .

الدرس الرابع عشر
النماذج (1)

سيكون هذا الدرس الأول من درسين سنتحدث فيهما عن النماذج وكيفية تضمينها في صفحات الويب.

مع أن النماذج تعتبر من المواضيع المتقدمة (وغير السهلة) نوعاً ما في لغة HTML إلا أن معظم مواقع الويب تكاد لا تخلو من وجودها، وذلك لعدة أسباب لعل من أهمها إيجاد إمكانية للتفاعل بين الموقع وصاحبه من جهة والزوار من جهة أخرى...أحيانا قد تحتاج كمصمم لموقع ويب أن تعرف آراء زوار موقعك في مسائل معينة وقد تكتفي برسائل البريد الإلكتروني التي يرسلوها لك، لكن عندما تريد معرفة أشياء محددة بذاتها فإن النماذج هي الخيار الأفضل لك. بالإضافة إلى إمكانية تنظيم البيانات المدخلة من خلالها وسهولة وسرعة استخدامها من قبل زوار الموقع. ومن أبرز الأمثلة على النماذج في مواقع الويب هي دفاتر الزوار وصفحات البحث عن الكلمات أو العبارات داخل المواقع.

لا تكمن صعوبة التعامل مع النماذج في كونها معقدة بحد ذاتها، كلا ... فهي إحدى العناصر التي تدعمها لغة HTML وهي مجرد وسوم عادية مثلها مثل الوسوم التي تعاملنا معها في جميع الدروس السابقة. وبإمكانك إنشاء النماذج في موقعك بنفس السهولة التي تدرج فيها جدولاً أو إطاراً (هذا بالطبع إذا كنت تعتقد أن الجداول والإطارات سهلة) لكن التداخل بينها (وأعني النماذج) وبين لغات البرمجة المتقدمة في الويب مثل JavaScript, CGI هي ما يجعلها تختلف عن سابقيها من الوسوم أو العناصر الأخرى. خاصة إذا احتجت إلى بعض المقاطع البرمجية من هذه اللغات ضمن نماذجك. أما إذا اكتفيت بالإمكانات المتواضعة التي توفرها HTML بالنسبة للنماذج. فما من مشكلة... لأنه سيكون بإمكانك التعامل معها بكل بساطة. وفي هذا الدرس لن نتطرق بالطبع إلى أي من اللغات سوى HTML.

ولنفترض الشكل التالي لدفتر زوار

كم شكلاً من أشكال إدخال البيانات يوجد في هذا الدفتر؟ الحقيقة أنه يوجد ستة أشكال هي كالتالي:



وهي الأشكال الموجودة في الدفتر فقط. وأود أن ألفت نظرك إلى وجود أشكال أخرى سوف يتم التعامل معها من خلال هذا الدرس.

والآن إلى العمل

مع أن الأشكال السابقة تختلف عن بعضها البعض من حيث المبدأ والمظهر (وطريقة التعريف أيضاً) إلا أنها يجب أن تدرج جميعاً ضمن وسمين أساسيين للنماذج هما:

</FORM> ... <FORM>

وكما جرت العادة نحتاج لتحديد بعض الخصائص التي تتعلق بطبيعة هذا النموذج. ولدينا هنا ثلاث خصائص:

ACTION

تحدد العنوان الذي سيتم إرسال بيانات النموذج إليه لتتم معالجتها بالصورة المطلوبة. وعادة يكون هذا عنواناً لبريد إلكتروني Email سوف يتم إرسال بيانات النموذج إليه. أو قد يكون عنواناً لبرنامج CGI موجود على الكمبيوتر الخادم Server الذي تتواجد عليه صفحة الويب، حيث يستقبل هذه البيانات ويعالجها حسب التعليمات الموجودة فيه كأن يضيفها مثلاً إلى إحدى الصفحات (كما يحدث عادة في دفاتر الزوار) أو يتحقق من صحة بعض الحقول المدخلة ومطابقتها لمعايير معينة، أو أن يقوم بالبحث عن كلمة أو عبارة ضمن صفحات الموقع كما في نماذج البحث الموجودة في مواقع الويب.

```
<FORM ACTION="mailto:someone@domain.com"> ... </FORM>
<FORM ACTION="name_and_address_of_CGI_script"> ... </FORM>
```

METHOD

تحدد الطريقة التي سيتم بها التعامل مع العنوان المحدد في الخاصية السابقة ACTION. وهناك قيمتين لهذه الخاصية هما: GET التي تستخدم في حالة كون عملية المعالجة داخلية أي تتم داخل الخادم Server نفسه. ففي مثالنا السابق عندما نستخدم نموذج البحث عن كلمة في الموقع، فإن عملية المعالجة (أي البحث) تجري مباشرة في الموقع. والقيمة الثانية هي Post وتستخدم عندما تكون عملية المعالجة خارجية كأن يتم إرسال البيانات إلى عنوان بريد إلكتروني.

```
<FORM ACTION="mailto:someone@domain.com" METHOD="post"> ... </FORM>
<FORM ACTION="name_and_address_of_CGI_script" METHOD="get"> ... </FORM>
```

ENCTYPE

هذه الخاصية تحدد طريقة الترميز التي سيتم إرسال البيانات وفقاً لها. وهي تأخذ القيمتين التاليتين: (يجب أن تكتب هذه القيم كما هي نصاً وحرفاً)

- application/x-www-form-urlencoded
- text/plain

وبدون الخوض في الأسباب التقنية التي أدت إلى إيجاد هذين النوعين من طرق الترميز أو في أمور برمجية بعيدة عن موضوعنا، فإن الدافع لإستخدام أي من القيمتين هو طبيعة عملية المعالجة التي ستجرى على البيانات أو طبيعة برنامج

البريد الإلكتروني الذي ستستقبل هذه البيانات من خلاله (إذا كان يدعم MIME أم لا، وهي إختصار للعبارة Multi-purpose Internet Mail Extentions وهي من المعايير السائدة في الإنترنت والتي تتعلق بنقل جميع أنواع البيانات من صوت وصورة وليس فقط النصوص من خلال البريد الإلكتروني). وما يعنينا هنا هو الفرق بين الطريقتين من حيث طريقة إرسال واستقبال البيانات. فعند استخدام text/plain ستصل البيانات بالشكل التالي:

```
NAME=Yahya Al-Sharif
Address=Nablus , Palestine
Email=yahya@palnet.com
```

(الكلمات Name, Email ,Address هي أسماء الحقول في النموذج ونقوم نحن بتعريفها أثناء عملية تصميم النموذج أما النصوص الظاهرة بعد إشارة المساواة فهي البيانات المدخلة، وسوف نتحدث عن تعريف أسماء الحقول بعد قليل)

أما عند استخدام application/x-www-form-urlencoded فستصل البيانات بالشكل:

```
NAME=Yahya+Al-Sharif&Address=Nablus+,+Palestine&Email=yahya@palnet.com
```

ولك أن تخيل مبلغ الصعوبة في تحليلها إذا احتوت على عشرات الحقول. لذلك تتوفر برامج خاصة تعرف بـِ Formaters تقوم بإعادة ترتيب البيانات المرسلة من خلال النماذج بشكل مفهوم بحيث تصبح كما لو كانت مرسلة بترميز text/plain لكن لا تعتقد أن الطريقة الأولى هي الأفضل دائماً فذلك يعتمد كما قلنا على طريقة المعالجة والنقل بالبريد. لذلك لا ضير من أن تجرب الطريقتين لتعرف أيهما أنسب لك.

إذن خلاصة القول: قد تكون أفضل صيغة لتعريف النموذج في حالة أردت استقبال البيانات من موقعك إلى عنوان بريدك الإلكتروني هي:

```
<FORM ACTION="mailto:email@domain.com" METHOD="post" ENCTYPE="text/plain">
...
</FORM>
```

وبهذا نكون قد إنتهينا من عملية تعريف النموذج وخصائصه، لكن انتظر فما زلنا في بداية الطريق.

نبدأ الآن في عملية تعريف أشكال البيانات في النموذج. ونستخدم الوسم <INPUT> لتعريفها والحقيقة أن هذه الأشكال هي مجرد خصائص أو بالأحرى قيم لخصائص تابعة لهذا الوسم. كيف؟ ... لنأخذ مثالاً على ذلك لأوضح لك هذا المفهوم

ملاحظة: إذا كنت تستخدم Sindbad 3.x فسيبدو الحقل والنص المجاور له بصورة معكوسة، وهي مشكلة ناتجة عن برنامج Netscape الذي يعمل من خلاله

Please enter your address: [Nablus, Palestine]

حسناً، لقد استخدمت الوسم <INPUT> لتعريف هذا الشكل (هذه إتفقنا عليها مسبقاً) ومن ثم قمت بإضافة الخاصية TYPE لهذا الوسم لتحديد نوع الشكل الذي أريده وأعطيتها القيمة TEXT أي

```
<FORM ...>
<INPUT TYPE="text">
</FORM>
```

لينتج لدينا هذا الشكل:

فقرة معترضة:

إليك جميع الأشكال (القيم) المستخدمة مع الخاصية TYPE وسوف أتركها الآن بدون تعليق لحين مناقشها لاحقاً بشكل مفصل. مع ملاحظة أن هناك شكلين آخرين ندرجهما بالوسوم <TEXTAREA> ,<SELECT>

<INPUT TYPE="text">	
<INPUT TYPE="password">	
<INPUT TYPE="hidden">	
<INPUT TYPE="radio">	
<INPUT TYPE="checkbox">	
<INPUT TYPE="submit">	Submit Query
<INPUT TYPE="reset">	
<INPUT TYPE="button">	

أرجو أن أكون قد وضحت لك الآن وظيفة الخاصية TYPE وجميع القيم المستخدمة معها

ونعود الآن إلى مثالنا.. الخاصية الثانية المستخدمة مع <INPUT> هي NAME وتستخدم لتسمية حقل البيانات حيث قمت بإعطاء الإسم address لهذا الحقل في المثال. (لك كل الحرية في إعطاء الإسم الذي تريده للحقل). والحقيقة أن هذا الإسم يعرّف الحقل في داخل النموذج نفسه، بحيث يمكن استخدامه فيما بعد للحاجات البرمجية وضرورات المعالجة إن وجدت من قبل البرامج التي قد تضيفها كمصمم للموقع. وحتى عندما تريد أن يُرسل النموذج إليك بالبريد فإن حقوله تعرّف بالاسم الذي أدرجته لها من خلال هذه الخاصية. (لاحظ ما قلتُه سابقاً عن تعريف أسماء الحقول عندما تحدثنا عن الترميز والطرق التي تصل بها محتويات النموذج). وكما ترى لا يوجد (حتى الآن) ما يدل على أن هذا الحقل يختص بإدخال العنوان.

```
<FORM ...>
<INPUT TYPE="text" NAME="address">
</FORM>
```

أما العبارة Please enter your address : فهي مجرد عبارة توضيحية أضفتها ليعرف الزائر ما الذي يجب عليه كتابته وتستطيع صياغة هذه العبارة كما تريد. ففي كل الأحوال ليس لها علاقة بجوهر النموذج نفسه بعكس الخاصية NAME.

```
<FORM ...>
Please enter your address : <INPUT TYPE="text" NAME="address">
</FORM>
```

Please enter your address :

أما بالنسبة للعبارة الظاهرة داخل الحقل Nablus, Palestine (أو أي عبارة أخرى تريدها) وهي بمثابة القيمة الإفتراضية التي تريدها للحقل، فبالإمكان إظهارها من خلال الخاصية VALUE. وهذه الخاصية تستخدم في الحالات التي نتوقع فيها كتابة قيمة دارجة أو متكررة من قبل معظم الزوار وللتسهيل عليهم يتم تعيينها كقيمة إفتراضية وبالطبع مع توفر إمكانية حذفها وكتابة ما يريدنه بدلاً منها.

```
<FORM ...>
Please enter your address : <INPUT TYPE="text" NAME="address" VALUE="Nablus, Palestine">
</FORM>
```

Please enter your address : Nablus, Palest

قد نحتاج أحياناً إلى تحديد حجم الحقل ولذلك نستخدم الخاصية SIZE مع الرقم الذي نريده كحجم للحقل، لنجرب الرقم 40

```
<FORM ...>
Please enter your address : <INPUT TYPE="text" NAME="address" VALUE="Nablus, Palestine" SIZE="40">
</FORM>
```

Please enter your address : Nablus, Palestine

أو لنجرب الرقم 10 أيضاً

Please enter your address : Nablus, Pal

لا يوجد للخاصية SIZE أي صفة تحكمية بالنسبة لحجم المدخلات التي يمكن للزائر أن يكتبها داخل الحقل. وبعبارة أخرى: صحيح أننا حددنا حجم الحقل لكن ذلك يسري فقط على مظهره على الشاشة. ولا يوجد ما يمنع الزائر من الكتابة بحيث يتجاوز النص حجم الحقل المحدد. وهنا يأتي دور الخاصية MAXLENGTH لتتحكم بالحد الأقصى للنص المدخل.

```
<FORM ...>
Please enter your address : <INPUT TYPE="text" NAME="address" VALUE="Nablus, Palestine"
SIZE="40" MAXLENGTH="30">
</FORM>
```

Please enter your address : Nablus, Palestine

حاول الكتابة في هذا الحقل لأكثر من 30 حرفاً وأنظر ماذا سيحدث؟

إنتهينا الآن من خصائص الوسم INPUT فما رأيك بإجمالها مرة أخرى؟ حسناً، هذه هي:

- TYPE: لتحديد نوع (شكل) حقل البيانات.

- NAME: لتعيين اسم لحقل البيانات.
- VALUE: لتعيين قيمة إفتراضية (مبدئية) لحقل البيانات.
- SIZE: لتحديد حجم حقل البيانات.
- MAXLENGTH: لتعيين الحد الأقصى لعدد الحروف المدخلة في الحقل.

النوع الثاني من الحقول المستخدمة في النماذج هو حقل password وهو يشبه الحقل text من حيث الخصائص تماماً غير أن مدخلاته تظهر على شكل ****** مهما كانت، وهو الفرق الوحيد بينهما. وربما تكون قد استنتجت الآن أن هذا النوع من الحقول يستخدم عندما يوجد حاجة لإدخال كلمة سر من قبل الزائر في النموذج

```
<FORM ...>
Please enter your name :
<INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40" MAXLENGTH="30">
Please enter your passwod :
<INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40" MAXLENGTH="30">
</FORM>
```

Please enter your name :

: Please enter your password

لاحظ أنني لم أرغب في كتابة قيم إفتراضية VALUES للحقول، ولذلك تركتها فارغة وأستطيع أيضاً أن ألغيها نهائياً من الشيفرة. وأنا في هذا المثال أردت أن أوضح لك عدم أهمية كتابة قيمة إفتراضية للحقول في بعض الحالات.

نأتي الآن إلى النوع الثالث من أنواع الحقول وهو hidden أي الحقل المخفي. وكما نستنتج من اسمه فهو لن يظهر ضمن النموذج. وهذا مثال:

```
<FORM ...>
Please enter your name :
<INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40" MAXLENGTH="30">

<INPUT TYPE="hidden" NAME="my forms" VALUE="form1">

Please enter your passwod :
<INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40" MAXLENGTH="30">
</FORM>
```

Please enter your name :

: Please enter your passwod

لاحظ هنا أن وجود هذا الحقل مثل عدمه بالنسبة لمظهر النموذج، وأن الزائر لن يتعامل معه بل وربما لن يعرف أن هناك حقلاً مخفياً. والسؤال هنا: ما الفائدة من وجود شيء مخفى لا إمكانية لاستخدامه؟ ولكي أجيب على هذا السؤال دعني أطرح لك مثالاً أو حالة قد تواجهك كمصمم صفحات ويب...
لنفرض أن لديك ثلاث صفحات تتضمن كل منها نموذجاً ما وأن هذه النماذج متشابهة. وتحتوي على نفس الحقول.
وعندما تصلك البيانات كيف ستستطيع تمييز أي من هذه النماذج استخدم لإرسال البيانات؟ بإمكانك إضافة هذا الحقل (الوهمي) وإسناد أي اسم وأي قيمة له في كل نموذج.

في النموذج الأول ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE="form1">
```

في النموذج الثاني ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE=" form2">
```

في النموذج الثالث ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE=" form3">
```

وبذلك عندما تصلك البيانات المرسلة من قبل أي زائر استخدم أي من النماذج الثلاثة سيصلك أيضا حقل إضافي قمت أنت نفسك بتعبئته سلفاً عندما صممت النموذج وذلك بأحد الأشكال التالية:

my forms=form1 أو
my forms=form2 أو
my forms=form3

إذن نستطيع القول أن الحقل المخفي هو لاستخدام المصمم وليس الزائر، وأن قيمته تدخل مباشرة عند التصميم.
ويستخدم بهدف تعريف قيم ما سيتم إرسالها جنباً إلى جنب ضمن بيانات النموذج التي قام الزائر بتعبئتها

ملاحظة مهمة بالنسبة للنماذج بشكل عام. من أجل إظهار النموذج بصورة مرتبة ومنسقة والتحكم بموقع الحقول فيه فمن الأفضل دائماً وضعه داخل جدول مع جعل الجدول بلا حدود.

```
<FORM ...>
<TABLE BORDER="0">
<TR>
<TD>Please enter your name : </TD>
<TD>
<INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40" MAXLENGTH="30">
</TD>
</TR>

<TR>
<TD>Please enter your password :</TD>
<TD>
<INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40" MAXLENGTH="30">
</TD>
</TR>
</TABLE>
</FORM>
```

وكما ترى تحتاج إلى القليل من العمل الإضافي لكنك بالمقابل ستحصل على النتيجة التالية

Please enter your name :	
Please enter your password :	

هكذا أفضل... أليس كذلك؟

الدرس الخامس عشر
النماذج (2)

وهو الدرس الثاني من درسين حول النماذج. لقد قمنا في الدرس السابق بمناقشة الوسوم الأساسية للنماذج وتعلمنا كيفية إدراج النماذج في صفحات الويب. كما قمنا بمناقشة بعض أشكال إدخال البيانات في النموذج وهي ,Text, Password Hidden هل تذكر كيف نقوم بتعريفها؟ راجع الدرس السابق إن أردت المزيد من التوضيح.

سوف نتابع الآن مع مجموعة من الأشكال الخاصة بالاختيار من متعدد وهي بالمناسبة ثلاثة أنواع: ,Radio Checkbox وقائمة الإختيار

نبدأ مع الشكل المسمى RADIO. ومن مسوغات استخدام هذا الشكل أن السؤال المطروح ينبغي أن يكون له إجابة واحدة فقط، أو بعبارة أخرى على الزائر أن يختار إجابة واحدة فقط.

وكمثال، لنفرض أنني أريد أن أسأل الزائر عن المتصفح الذي يستخدمه (كما هو موجود في دفتر الزوار على شكل قائمة إختيار) لكن بدلا من أن يكون على شكل قائمة إختيار أريده أن يكون على شكل RADIO وذلك بالشكل التالي:(أود أن أذكرك أن عناصر النموذج تظهر بشكل معكوس إذا كنت تستخدم Sindbad 3.0)

Sindbad 3.0

Sindbad 4.0

Ms Explorer 3.0

Ms Explorer 4.0

فكيف ننشيء مثل هذه القائمة؟ ... حسناً، لنبدأ من الصفر ونعرّف نموذجاً

```
<FORM>

</FORM>
```

ثم لنقم بتعريف هذا الشكل، هل تذكر الوسم الخاص بذلك؟ إنه <INPUT>

```
<FORM>
<INPUT TYPE="radio">
</FORM>
```

لكن بما أن هناك أربعة مدخلات، إذن نحتاج إلى أربعة وسوم

```
<FORM>
<INPUT TYPE="radio"> <BR>
<INPUT TYPE="radio"> <BR>
<INPUT TYPE="radio"> <BR>
<INPUT TYPE="radio"> <BR>
</FORM>
```

نحتاج الآن إلى تسمية هذه المدخلات، أي أننا سنستخدم الخاصية NAME معها. أما الاسم المعطى بحد ذاته فمن الأفضل أن يكون مرتبطاً نوعاً ما بموضوع السؤال، ليس لأن هذا ضروري للنموذج بل هو ضروري لك كشخص سيقوم باستقبال البيانات المرسلة من خلال النموذج، وبالتالي من الأفضل أن يوجد عنوان معبّر للبيانات بغرض التمييز. وبما أننا هنا نتحدث عن المتصفحات فليكن هذا الاسم هو browser

```
<FORM>
<INPUT TYPE="radio" NAME="browser"> <BR>
<INPUT TYPE="radio" NAME="browser"> <BR>
<INPUT TYPE="radio" NAME="browser"> <BR>
<INPUT TYPE="radio" NAME="browser"> <BR>
</FORM>
```

وكما تلاحظ من النتيجة أن هذه التسمية هي ضمنية فقط ولا تؤثر على شكل النموذج (راجع الدرس السابق) لكن أي إختيار سيقوم به الزائر من هذه الأربعة خيارات سوف يصلك تحت الاسم browser.

الخطوة التالية هي إعطاء قيمة لكل مدخلة في هذه القائمة وذلك حسب ما نراه مناسباً، إذن حان الوقت لاستخدام الخاصية VALUE:

```
<FORM>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind3"> <BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind4"> <BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Msie3"> <BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Msie4"> <BR>
</FORM>
```

وهنا أيضا نلاحظ أن لا تغيّر في شكل النموذج ظاهرياً مع إضافة هذه الخاصية. لكن مع ذلك فقد قمنا حتى الآن بتسمية الحقول وإعطاء كل حقل قيمة محددة. وفعلياً لقد إنتهينا من هذا النموذج. لكن بالطبع نحن لا نتوقع أن يكون الزائر عالماً بالغيب لكي يخمن أي من هذه الحقول تختص بكل قيمة. لذلك بقى علينا تعريف كل حقل باسم صريح يوضح محتواه.

```
<FORM>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind3"> Sindbad 3.0 <BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind4"> Sindbad 4.0<BR>
```

```
<INPUT TYPE="radio" NAME="browser" VALUE="Msie3"> MS Explorer 3.0<BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Msie4"> MS Explorer 4.0<BR>
</FORM>
```

○ Sindbad 3.0

○ Sindbad 4.0

○ MS Explorer 3.0

○ MS Explorer 4.0

وهناك خاصية تتعلق بهذا النوع من الحقول، وهي أنك إذا أردت أن يظهر أحدها وقد تم اختياره بشكل تلقائي فعليك بإضافة الخاصية CHECKED إليه، مع ترك كل الحرية للزائر في اختيار ما يريده فيما بعد.

```
<FORM>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind3"> Sindbad 3.0 <BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind4" CHECKED>
Sindbad 4.0<BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Msie3"> MS Explorer 3.0<BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Msie4"> MS Explorer 4.0<BR>
</FORM>
```

○ Sindbad 3.0

◉ Sindbad 4.0

○ MS Explorer 3.0

○ MS Explorer 4.0

وأخيراً... أود أن أوضح لك الصورة التي يصلك بها النموذج عند اختيار أحد حقوله (ولنفترض أنه الخيار الثالث). وهي بالشكل التالي:

browser=Msie3

نأتي الآن إلى الشكل الثاني من أشكال الإختيار من متعدد والذي يدعى CHECKBOX. ظاهرياً لا يختلف هذا الشكل عن الشكل الذي سبقه، لكن عملياً هناك اختلافات جذرية من حيث المفهوم والتعريف. وأنا أفضّل أن نبقى على استخدامنا للمثال السابق حتى يسهل علينا تمييز الفروق.

☐ Sindbad 3.0

☐ Sindbad 4.0

☐ MS Explorer 3.0

☐ MS Explorer 4.0

قبل أن نستمر لو قتم بالنقر على أكثر من حقل في القائمة السابقة فإن باستطاعتك اختيار أكثر من حقل في نفس الوقت! وهذا هو الفرق الأول بين CHECKBOX و RADIO ففي RADIO يمكن اختيار حقل واحد فقط ليس أكثر.
لنقم الآن بتعريف هذه الحقول، وتسميتها بشكل مباشر ومن ثم سنعلّق عليها:

```
<FORM>
<INPUT TYPE="checkbox" NAME="Sind3" VALUE="Yes"> Sindbad 3.0 <BR>
<INPUT TYPE="checkbox" NAME="Sind4" VALUE="Yes"> Sindbad 4.0 <BR>
<INPUT TYPE="checkbox" NAME="Msie3" VALUE="Yes"> MS Explorer 3.0 <BR>
<INPUT TYPE="checkbox" NAME="Msie4" VALUE="Yes"> MS Explorer 4.0 <BR>
</FORM>
```

ماذا تلاحظ؟ أولاً لقد أسندنا القيمة checkbox للخاصية TYPE. ثم أعطينا لكل حقل في القائمة إسماً مميزاً في الخاصية NAME يختلف عن باقي الحقول. أما الخاصية VALUE فأعطيناها قيمة موحدة لجميع الحقول. وبالطبع قمنا في النهاية بكتابة الأسماء التعريفية لكل حقل.

> في RADIO نستطيع اختيار حقل واحد فقط أما في CHECKBOX فنختار أكثر من حقل، لذلك يستخدم عادة في الحالات التي يحتمل أن نحصل فيها على عدة أجوبة لنفس السؤال.
>
> في RADIO تكون أسماء الحقول موحدة والقيم مختلفة، أما في CHECKBOX فتكون الأسماء مختلفة والقيم موحدة

كيف ستصل البيانات؟ حسناً لنفرض أنه تم اختيار الحقلين الثاني والرابع فسوف تصلك النتيجة بالشكل التالي:

Sind4=Yes
Msie4=Yes

كما نستطيع أيضاً تعليم بعض الحقول بصورة تلقائية كما فعلنا مع RADIO باستخدام نفس الخاصية CHECKED

```
<FORM>
<INPUT TYPE="checkbox" NAME="Sind3" VALUE="Yes" CHECKED>
Sindbad 3.0 <BR>
<INPUT TYPE="checkbox" NAME="Sind4" VALUE="Yes"> Sindbad 4.0 <BR>
<INPUT TYPE="checkbox" NAME="Msie3" VALUE="Yes" CHECKED>
MS Explorer 3.0 <BR>
<INPUT TYPE="checkbox" NAME="Msie4" VALUE="Yes"> MS Explorer 4.0 <BR>
</FORM>
```

- [x] Sindbad 3.0
- [ ] Sindbad 4.0
- [x] MS Explorer 3.0
- [ ] MS Explorer 4.0

النوع الثالث من أشكال الإختيار من متعدد هو قوائم الاختيار، وهذا النوع سوف يقودنا إلى وسوم جديدة من وسوم التعريف والتي ستستخدم بدلاً من <INPUT> وهي

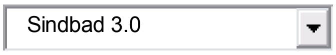
```
<SELECT>
   <OPTION>
   <OPTION>
   <OPTION>
```

```
  .....
  .....
</SELECT>
```

بحيث أن <SELECT> ... </SELECT> تحددان بداية ونهاية القائمة، والوسم <OPTION> الذي يوضع دائما بينهما يستخدم لتحديد كل عنصر من عناصر القائمة. لنعد إلى مثالنا السابق لنرى كيف يمكن وضع الخيارات في قائمة

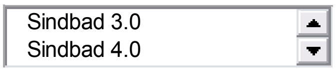
```
<FORM>
<SELECT>
  <OPTION> Sindbad 3.0
  <OPTION> Sindbad 4.0
  <OPTION> MS Explorer 3.0
  <OPTION> MS Explorer 4.0
</SELECT>
</FORM>
```

وبذلك تكون النتيجة هي:



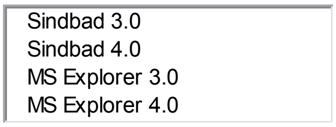
وكالمعتاد لا يخلو الأمر من وجود خصائص تحدد طريقة عمل هذه الوسوم. وهناك خصائص مشتركة عرفناها في الأشكال السابقة سيتم استخدامها هنا أيضاً كما يوجد خصائص جديدة تتعلق فقط بهذا الشكل من حقول البيانات. فبالنسبة لـ <SELECT> يوجد الخاصية NAME وهي كما تعلم تحدد اسم القائمة. كما توجد الخاصية SIZE التي تحدد حجم (أو بالأحرى) ارتفاع القائمة، وبالتالي عدد البيانات الظاهرة فيها. وهي تأخذ أي قيمة عددية صحيحة.

```
<FORM>
<SELECT NAME="browser" SIZE="2">
  <OPTION> Sindbad 3.0
  <OPTION> Sindbad 4.0
  <OPTION> MS Explorer 3.0
  <OPTION> MS Explorer 4.0
</SELECT>
</FORM>
```



وطالما بالإمكان عرض القائمة بأي ارتفاع نريد، وقد يصل إلى حد عرض جميع بيانات القائمة معاً، فإن هناك إمكانية أيضاً لجعل اختيار البيانات من هذه القائمة متعدداً وليس فقط قيمة واحدة، كيف؟؟ بإضافة الخاصية MULTIPLE. لنقم الآن بعرض جميع القيم (لدينا أربعة قيم، إذن القيمة المكتوبة مع SIZE يجب أن تكون 4)، ومن ثم لنتح المجال أمام الزائر لاختيار أكثر من قيمة واحدة في القائمة.

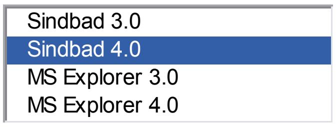
```
<FORM>
<SELECT NAME="browser" SIZE="4" MULTIPLE>
  <OPTION> Sindbad 3.0
  <OPTION> Sindbad 4.0
  <OPTION> MS Explorer 3.0
  <OPTION> MS Explorer 4.0
</SELECT>
</FORM>
```



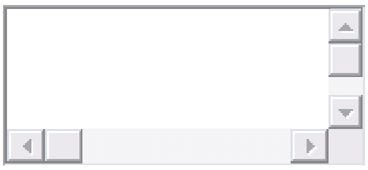
لاحظ أنه لأداء عدة اختيارات يجب أن تقوم بالضغط على المفتاح ctrl بصورة متواصلة أثناء عملية الإختيار.

أما الخصائص المستخدمة مع الوسم <OPTION> فهي VALUE والتي استخدمناها من قبل وسنستخدمها الآن لإعطاء قيمة لكل حقل بيانات في القائمة. وكذلك الخاصية SELECTED والتي نكتبها مع أي <OPTION> نريد أن يظهر وقد تم اختياره بصورة تلقائية.

```
<FORM>
<SELECT NAME="browser" SIZE="4" MULTIPLE>
   <OPTION VALUE="Sindbad 3.0"> Sindbad 3.0
   <OPTION VALUE="Sindbad 4.0" SELECTED> Sindbad 4.0
   <OPTION VALUE="MS Explorer 3.0"> MS Explorer 3.0
   <OPTION VALUE="MS Explorer 4.0"> MS Explorer 4.0
</SELECT>
</FORM>
```



الشكل التالي من أشكال حقول البيانات يدعى TEXTAREA

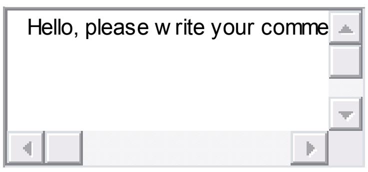


و هو المستخدم عادة لكتابة التعليقات الحرة في النموذج ويتم إدراجه بكتابة الوسوم

<TEXTAREA> ... </TEXTAREA>

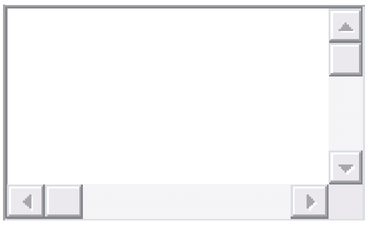
هل تستطيع تخمين الخصائص المستخدمة معه؟ بالطبع لا بد من وجود الخاصية NAME لإعطاءه اسم التعريف. لكن لا وجود للخاصية VALUE ، وبالمقابل فإن أي نص يكتب بين الوسمين سيتم عرضه داخل الحقل بصورة تلقائية

```
<TEXTAREA NAME="comments">
Hello, please write your comments here :-)
</TEXTAREA>
```



كما توجد خصائص لتحديد مساحة هذا الحقل عرضاً وارتفاعاً، وهي COLS التي تحدد العرض و ROWS التي تحدد الإرتفاع

```
<TEXTAREA NAME="comments" COLS="30" ROWS="6">
</TEXTAREA>
```



أما الخاصية الأخيرة هنا فهي WRAP التي تحدد طريقة إلتفاف النص المكتوب داخل الحقل (لا تعمل هذه الخاصية مع MS Explorer 3.0) وهناك ثلاثة قيم تأخذها وهي على النحو التالي:

virtual : التي تعني أن النص سيلتف على عدة أسطر عند كتابته ولكنه سيصلك عند إرساله على شكل سطر واحد متتابع (حاول الكتابة داخل الحقول وأنظر كيفية تأثير هذه الخاصية على كل منها)

```
<TEXTAREA NAME="comments" COLS="30" ROWS="6" WRAP="virtual">
</TEXTAREA>
```



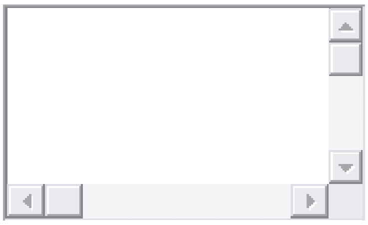
physical : تعني أن النص سيلتف على عدة أسطر وسيصلك أيضاً على هذا النحو عند إرساله

```
<TEXTAREA NAME="comments" COLS="30" ROWS="6" WRAP="physical">
</TEXTAREA>
```



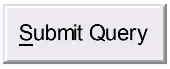
off : تعني أن النص لن يلتف بصورة تلقائية على عدة أسطر لكنه على أية حال سيصلك بنفس الشكل الذي تم إدخاله به

```
<TEXTAREA NAME="comments" COLS="30" ROWS="6" WRAP="off">
</TEXTAREA>
```



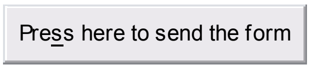
حسناً، بافتراض أننا إنتهينا من كتابة الشيفرة الخاصة بالنموذج وننتظر من أي زائر للموقع أن يملأه، والسؤال هو كيف يمكن له أن يرسله فعلياً؟ نعود الآن إلى الوسم <INPUT> وهذه المرة مع النوع submit والتي ستقوم تلقائياً بإنشاء زر سيقوم عند النقر عليه بإرسال البيانات التي تم ملؤها في النموذج.

```
<INPUT TYPE="submit">
```



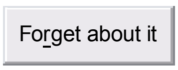
لاحظ أن Submit أو (Submit Query في Netscape) ظاهرة على الزر وهي العبارة الإفتراضية، فإذا أردت تغييرها فعليك باستخدام الخاصية VALUE لهذا الغرض

```
<INPUT TYPE="submit" VALUE="Press here to send the form">
```



في حالة كان زوار موقعك من النوعية المترددة من الناس والذين قد يغيرون آرائهم في آخر لحظة، يمكنك أن تتيح لهم إمكانية مسح ما كتبوه في النموذج وإلغاء الأمر، وذلك باستخدام reset كنوع TYPE للوسم <INPUT> بنفس طريقة التعريف والخصائص المستخدمة مع submit.

```
<INPUT TYPE="reset" VALUE="Forget about it">
```



والذي يقوم بإنشاء زر button> هو INPUTالشكل الأخير من أشكال البيانات في النماذج والمدرج مع الوسم < ضمن النموذج، وهو مرتبط بالنماذج التي تحتوي على نصوص برمجية (أو برامج مكتملة) من لغات متقدمة مثل JavaScript كونه يستخدم لتشغيل هذه البرامج وإطلاقها. وطبعاً هناك طرق معينة لربطها مع البرامج وليس هنا المجال لطرحها. لكن مبدئياً أقول إن طريقة الإدراج والتعريف هي ذاتها المستخدمة مع reset, submit.

```
<INPUT TYPE="button" VALUE="This is a sample button">
```

وأخيراً ... وصلنا الآن إلى نهاية حديثنا عن النماذج. فما رأيك؟ هل هو من المواضيع السهلة أم الصعبة؟ لا شيء سهل في بدايته. لذلك من الأفضل لك أن تحاول دائماً التدرب أولاً بأول على الوسوم المشروحة، بل والعودة إلى الدروس السابقة إذا اقتضى الأمر وخاصة إذا تداخلت بعضها مع الدروس الأحدث.

الدرس السادس عشر
المتصفحات والوسوم الخاصة

سنناقش هنا واحداً من المواضيع المهمة.. والشائكة.. والمزعجة نوعاً ما في هذه اللغة، وفي مجال تصميم صفحات الويب عموماً. وهو موضوع الوسوم الخاصة... فما هي هذه الوسوم؟ ... حسناً، إليك القصة:-

هل سبق لك وأن صادفت عبارة *Best Viewed with Netscape Navigator* أو *Best Viewed with MS Explorer Internet* أو (أفضل مشاهدة بواسطة المتصفح نيتسكيب ... أو إكسبلورر) عند زيارتك لبعض مواقع الإنترنت؟ حتماً لقد شاهدت مثل هذه العبارات تذيل الكثير من المواقع. والحقيقة أن مصممي هذه المواقع ليسوا بالضرورة من المعجبين بهذا المتصفح أو ذاك بحيث يعلنوا ولائهم له دون غيره من المتصفحات. لكن بكل بساطة ربما قاموا بتضمين بعض الوسوم أو الخصائص التي يدعمها ذلك المتصفح فقط في صفحاتهم...! بحيث إذا قام زائر ما بالدخول إلى ذلك الموقع مستخدماً متصفحاً آخر غير الموصى به فلن يشاهد نفس النتيجة التي يشاهدها زائر آخر يستخدم المتصفح المطلوب. (طبعاً لا أستطيع أن أعمم هذا لأن هناك العديد من المصممين الذين يفضلون فعلاً متصفحاً عن غيره سواءً قاموا بتضمين الوسوم الخاصة به أم لا)

نعم، يوجد هناك العديد من الوسوم في لغة HTML التي قامت الشركات المنتجة للمتصفحات كمايكروسوفت ونيتسكيب بتطويرها بحيث تعمل على متصفحها الخاص دون غيره من المتصفحات Browser Specific Tags (من قال لك أن الشركات الكبرى تتعامل مع بعضها البعض كالكبار). وقد يصل الأمر أحياناً إلى وجود وسوم موحدة تدعمها جميع المتصفحات لكن طريقة تنفيذ وعرض هذه الوسوم هي التي تختلف. وأذكرك بما قلناه في الدرس الرابع عند الحديث عن القوائم. فقد شاهدنا كيف أن كلا المتصفحين يدعمان الوسم <UL> والخاصية TYPE لكنهما يختلفان على القيم التي تدرج معهما. وهذا مثال واحد فقط على هذا الإختلاف.

وأكثر من ذلك، لا ننسى أن شركات البرامج تقوم بشكل مستمر بعمليات التحديث والتحسين لبرامجها وإطلاق إصدارات جديدة لها بين الحين والآخر، لذلك من غير المستبعد أن يختلف الدعم للوسوم والخصائص من قبل نفس المتصفح بإصداراته المختلفة، وبالتالي اختلاف مظهر الصفحات. وهذه المسألة بالذات (وأعني اختلاف مظهر الصفحات) هي ما تسبب أرقاً لمصممي صفحات الويب. والسؤال الملّح دوماً لديهم هو: كيف أحصل على صفحة ويب لا تختلف في عرضها من متصفح لآخر؟ وقد يكون جوابه أكثر إلحاحاً، إذ يكون من الصعب أحياناً تحقيق هذا الهدف. وتزداد صعوبته كلما ازدادت العناصر والتنسيقات المختلفة التي تحتويها الصفحة. بينما يسهل التحكم في صفحة تقتصر محتوياتها على النصوص فقط ويكون من الممكن إخراجها بنفس الطريقة في جميع المتصفحات.

فما الحل إذن؟

مع أن بعض هذه الوسوم الخاصة تعطي تأثيرات رائعة للصفحة، وتضفي عليها مظهراً جميلاً. لكنك بالمقابل لا تضمن أن جميع زوار موقعك يستخدمون نفس المتصفح، فهذا مستحيل بالطبع. لذلك لا أقول لك لا تستخدم هذه الوسوم لكن حاول قدر الإمكان تجنبها إذا أردت أن تظهر صفحتك بنفس الطريقة التي تريدها لجميع الزوار. كذلك حاول دائماً استعراض صفحاتك باستخدام المتصفحين وباستبانات مختلفة للشاشة فهذا يعطيك فكرة مبدئية عن الطريقة التي سيشاهدها بها زوارك على مختلف المتصفحات.

> أنا هنا أتحدث عن الصفحات الإنجليزية القياسية عموماً، أما الصفحات العربية فهي حكاية أخرى لأن الدعم الموجود للغة العربية في المتصفحات ليس قياسياً نتيجة لعدة أسباب لا مجال لذكرها في هذا المقام. فمثلاً متصفح نيتسكيب لا يدعمها أصلاً ويحتاج إلى برنامج إضافي Plugin من شركة صخر وهو سندباد. وهناك متصفح آخر يدعم العديد من اللغات العالمية بما فيها العربية، وهو

تانجو من شركة أليس. لكن أنا شخصياً أفضل مايكروسوفت إكسبلورر ليس من ناحية استخدامي لأية وسوم خاصة به بل لأنه الأفضل في عرض النصوص العربية والصفحات ثنائية اللغة أكثر من غيره من المتصفحات.

قبل أن نبدأ باستعراض الوسوم لا بد لي من التنويه بأني استخدم برنامجي Sindbad 3.x و Ms Internet Explorer 3.x وعندما أتحدث عن الوسوم والخصائص وتوافقها مع المتصفحات فأنا أعني هذين المتصفحين وأعني رقم الإصدار أيضاً (3.x). وذلك لأنهما لا زالا الأكثر رواجاً (حتى تاريخ كتابة هذا الدرس)، سواء بيننا كمستخدمين عرب أو بشكل عام بين مستخدمي الإنترنت عالمياً. ومن المحتمل إذا كنت تستخدم إصداراً أحدث لأحد هذه البرامج أن تلاحظ أنه قد أصبح يدعم الوسوم الخاصة بالبرنامج الآخر (وهذا بالطبع ليس ذنبي، بل ذنب أولئك الذين يطورون برامجهم بشكل دائم ومتسارع ولا يتركوننا نلتقط أنفاسنا بين كل إصدار وآخر). وعلى أية حال إذا كنت ممن يقبلون النصيحة المجانية، حاول دائماً تصميم صفحاتك مع افتراض أسوأ الإحتمالات أي واضعاً نصب عينيك أن زوار موقعك يستخدمون متصفحات قديمة. وبذلك تضمن أن معظمهم إن لم يكن جميعهم سيشاهدون موقعك بنفس الصورة وبدون أي مشاكل تتعلق بالتوافقية بين المتصفحات والوسوم.

وبالنسبة لطريقة عرض الوسوم لهذا الدرس، فنظراً لخصوصيتها سوف أتبع هنا منهجاً مختلفاً عن الدروس السابقة يقوم على عرض الوسوم الخاصة بكل متصفح في صفحة منفصلة. وذلك بهدف حصر الوسوم المخصصة لكل متصفح وعدم تداخلها مع الوسوم الأخرى. وبالتالي تجنب احتمال حدوث خلل في بنية الصفحات عند عرضها من خلال متصفح لا يدعم بعضاً من هذه الوسوم والشيفرات الخاصة بها.
لكن هذا لا يعني عدم قدرتك على مشاهدة الصفحة الخاصة بوسوم أحد المتصفحات إذا كنت تستخدم الآخر. بالطبع سوف يكون بإمكانك مشاهدة النص المكتوب والشيفرات المستخدمة كالمعتاد (فنحن في النهاية لا زلنا نستخدم لغة HTML) لكن كل ما هنالك أنه لن يكون بإمكانك مشاهدة تأثيرها إلا إذا استخدمت المتصفح المناسب.

الوسوم الخاصة

Navigator Netscape

ملاحظة مهمة: تحتوي هذه الصفحة على وسوم خاصة بالمتصفح Netscape Navigator 3.x فإذا كان متصفحك خلاف ذلك فلن تستطيع مشاهدة الصفحة بالصورة المناسبة

- نص وامض <BLINK>
- اعمدة متعددة <MULTICOL>
- أصوات موسيقية ومقاطع فيديو <EMBED>

</BLINK> ... <BLINK>

ضع أي نص تريده بين هذين الوسمين و سوف تشاهده يومض أمامك على الشاشة

```
<BLINK>H.P in arabic </BLINK>
```

H.P in arabic

</MULTICOL> ... <MULTICOL>

للصحفيين فقط... هل تريد أن يظهر النص في صفحاتك مقسماً إلى عدة أعمدة كما هو الحال في الجرائد والمجلات؟ هذا ما توفره لك هذه الوسوم. فلك أن تحدد عدد الأعمدة التي تريدها ومن ثم مقدار المسافة الفارغة التي تفصل بين كل عمود والعمود الذي يجاوره وكذلك العرض الذي تشغله الأعمدة ككل.

لنجرب وضع فقرة في عمودين. وسوف نستخدم لهذا الغرض الخاصية COLS لتقوم بتحديد عدد الأعمدة المطلوبة

```
<MULTICOL COLS="2">
OUR PARAGRAPH IS HERE
</MULTICOL>
```

والآن سنستخدم الخاصية GUTTER لتحديد المسافة الفاصلة بين كل عمود وآخر بالبيكسل.

```
<MULTICOL COLS="2" GUTTER="70">
OUR PARAGRAPH IS HERE
</MULTICOL>
```

لا تتوفر الإمكانية في هذه الوسوم لتحديد عرض كل عمود على حدة لكن بالمقابل نستطيع تحديد عرض جميع الأعمدة معاً (أي النص ككل) وذلك إما بوضع قيمة مطلقة وثابتة أو بوضع قيمة نسبية تؤدي إلى إحداث تغير في عرض النص تبعاً لعرض نافذة المتصفح. والخاصية التي تؤدي هذا الغرض هي WIDTH.

سأوضح لك هذه الخاصية من خلال المثالين التاليين: سأقوم أولاً بتحديد عرض النص بالقيمة الثابتة 500، ثم استخدام قيمة نسبية هي 60%

```
<MULTICOL COLS="2" GUTTER="70" WIDTH="500">
OUR PARAGRAPH IS HERE
</MULTICOL>


<MULTICOL COLS="2" GUTTER="70" WIDTH="60%">
OUR PARAGRAPH IS HERE
</MULTICOL>
```

<EMBED>

يدعم Netscape هذا الوسم <EMBED> ، والذي يستخدم عموماً لإدراج ملفات الملتيميديا في صفحات الويب سواءً كانت مقاطع موسيقية أو ملفات فيديو. لكن يشترط هنا وجود برامج إضافية مشغّلة لهذه الملفات تدعى Plugins. يتمّ إلحاقها بـِ Netscape لأنه لا يحتوي أصلاً إلا على بعض المشغلات المدمجة فيه بينما يفتقر إلى مشغلات أخرى. (لمعرفة المشغلات المثبتة في برنامجك أكتب about:plugins في المستطيل الخاص بالعناوين في متصفحك ثم أضغط Enter أو اختر الأمر About Plug-ins من قائمة Help) وعادة يتم الحصول على هذه المشغلات من الإنترنت. كذلك يحفل موقع شركة Netscape بالعديد من هذه المشغلات أو على الأقل الوصلات التشعبية التي تقودك إليها.



إن المشغل المدمج الخاص بملفات mid هو الذي أظهر الصورة الموجودة في الأعلى والتي تستخدم من أجل التحكم في الملف الصوتي. والحقيقة أنه يوجد لكل مشغّل شاشة التحكم الخاصة به والتي تظهر بصورة شبيهة لما تراه الآن.

من الخصائص المستخدمة مع هذا الوسم ما يلي:

SRC: لتحديد مصدر الملف المراد تشغيله

```
<EMBED SRC="onceupon.mid">
```

لاحظ أن شاشة التحكم قد ظهرت بشكل مشوه، فهي مجرد صورة مقصوصة. ونحن نعرف أننا نستطيع التحكم بحجم الصور من خلال الخصائص HEIGHT, WIDTH إذن...

```
<EMBED SRC="onceupon.mid" WIDTH="145" HEIGHT="61">
```

AUTOSTART: تحدد ما إذا كان الملف سيتم تشغيله تلقائياً عند تحميل الصفحة. وتأخذ القيم true للتشغيل التلقائي (كما في هذه الصفحة) و false لعدم التشغيل، وهي القيمة الإفتراضية. وكما ترى تظهر في هذه الحالة شاشة التحكم فقط دون الصوت، بما يتيح للزائر التحكم في التشغيل أو عدمه عندما يريد ذلك.

والخاصية الأخيرة هي LOOP التي تحدد إذا كان الملف سيتم عرضه بشكل متكرر أم لا وتأخذ القيم true, false. لاحظ أن لا إمكانية هنا لتحديد عدد مرات التكرار كما توفره الوسوم الخاصة بـ MS Explorer.

الوسوم الخاصة 1

Internet Explorer MS

ملاحظة مهمة: تحتوي هذه الصفحة على وسوم خاصة بالمتصفح MS Internet Explorer فإذا كان متصفحك خلاف ذلك فلن تستطيع مشاهدة الصفحة بالصورة المناسبة

- أصوات موسيقية <BGSOUND>
- لافتات <MARQUEE>

<BGSOUND>

تستطيع تعيين أي ملف صوتي يحمل الإمتداد mid أو wav أو au بحيث يعمل تلقائياً عندما يتم تحميل الصفحة في المتصفح، وذلك باستخدام الوسم <BGSOUND> (أي BackGround Sound) فعلى سبيل المثال، قمت بإدراج مقطع والمسمى baladi.mid بالطريقة التالية

```
<BGSOUND SRC="baladi.mid">
```

إذن هناك الخاصية SRC والتي تستخدم كما ترى لتحديد اسم الملف المدرج. كما يوجد أيضاً الخاصية LOOP التي تحدد عدد مرات تكرار عزف المقطوعة أي أنها تأخذ أرقاماً صحيحة عادة. أما لو قمت بإعطاءها القيمة 1- أو القيمة infinite فهذا سوف يؤدي إلى تكرار عزف المقطوعة إلى ما لا نهاية (لا تخف، فهذا لا يعني أن العزف سيستمر بعد أن تقوم بإطفاء جهازك) فقط سوف تسمعها طالما أنك موجود في الصفحة وسيتوقف بمجرد انتقالك لصفحة أخرى أو خروجك من البرنامج.

```
<BGSOUND SRC="baladi.mid" LOOP="infinite">
<BGSOUND SRC="baladi.mid" LOOP="3">
```

أما في حالة أردت إدراج وصلات تشعبية لملفات صوتية في موقعك بحيث يمكن للزائر أن يحمّلها أو أن ينقر عليها إذا أراد سماعها، فذلك يتم بالطريقة الإعتيادية لإدراج الوصلات والتي تعلمناها في الدرس السادس أي كما يلي:

```
<A HREF="baladi.mid"> ... </A>
```

وهي -كما تعلم- طريقة عامة ليس لها علاقة بنوع المتصفح المستخدم

One Man's Dream   Upon A Time Once

</MARQUEE> ... <MARQUEE>

(نص يتحرك من اليمين الى اليسار)

ما رأيك بهذا النص المتسكع أمامك على الشاشة؟ جميل ..أليس كذلك؟ هذه اللافتة (إن جاز التعبير) هي إحدى المؤثرات الخاصة التي يمكن إحداثها في MS Explorer. بواسطة هذه الوسوم. وبكل بساطة هذه هي الشيفرة الخاصة بذلك

```
<MARQUEE>
H.P in arabic
</MARQUEE>
```

بالطبع لا يمكن أن تمر الأمور هنا ببساطة فلا بد من وجود خصائص لهذه الوسوم، والحقيقة أن هناك إحدى عشرة خاصية نستخدمها (عدد ليس بالقليل بالنسبة لوسوم خاصة). لكنها على أية حال ليست صعبة التطبيق بل إننا تعاملنا بشكل أو بآخر معها مسبقاً وخاصة مع الوسم <IMG> وهذه هي الخصائص أسردها لك مع الأمثلة:

BGCOLOR: لتحديد لون الخلفية

```
<MARQUEE BGCOLOR="#FF0000">
H.P in arabic
</MARQUEE>
```

HEIGHT: لتحديد إرتفاع اللافتة، وهي إما أن تأخذ قيمة ثابتة أو قيمة نسبية حسب إرتفاع نافذة المتصفح.

```
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80">
H.P in arabic
</MARQUEE>
```

WIDTH: لتحديد عرض اللافتة، وهنا أيضاً إما أن تأخذ قيمة ثابتة أو نسبية

```
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80" WIDTH="40%">
H.P in arabic
</MARQUEE>
```

ALIGN: لتحديد المحاذاة العمودية للنص الذي قد يتواجد على جانبي اللافتة، وهي تأخذ القيم top, middle و bottom وهي الإفتراضية أي التي تطبق تلقائياً عند عدم إدراج هذه الخاصية.

```
Welcome to
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80" WIDTH="40%">
H.P in arabic
</MARQUEE>
Have a good time.
```

.good time Have a Welcome to

```
Welcome to
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80" WIDTH="40%" ALIGN="middle">
H.P in arabic
</MARQUEE>
Have a good time.
```

.Have a good time Welcome to

```
Welcome to
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80" WIDTH="40%" ALIGN="top">
H.P in arabic
</MARQUEE>
Have a good time.
```

.Have a good time Welcome to

HSPACE: لتحديد المسافة الفارغة التي تفصل بين اللافتة وبين النص الموجود على جانبيها

```
Welcome to
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80" WIDTH="40%" HSPACE="30">
H.P in arabic
</MARQUEE>
Have a good time.
```

.Have a good time Welcome to

VSPACE: تعمل مثل الخاصية السابقة لكنها تحدد المسافة الفارغة عمودياً من الأعلى والأسفل

حتى الآن كانت الخصائص التي ناقشناها مجرد خصائص مظهرية تتعلق بشكل ومظهر اللافتة دون تغيير طريقة عملها. والآن نأتي إلى الخصائص الفنية:

BEHAVIOR تحدد سلوك النص الموجود في اللافتة وطريقة عمله وهناك ثلاثة قيم (أو بالأحرى) ثلاثة أساليب لحركة النص وهي التالية:

- scroll يتحرك بنفس الإتجاه من جانب إلى آخر وبصورة مستمرة وهي القيمة الإفتراضية، لذلك لا يهم إن كتبت أم لا.
- slide يتحرك النص مرة واحدة من جانب إلى الجانب الآخر ويتوقف عنده.

```
<MARQUEE BGCOLOR="#FF0000" WIDTH="80%" BEHAVIOR="slide">
H.P in arabic
</MARQUEE>
```

من المحتمل أن يكون النص في المثال أعلاه متوقفاً بعد أن يكون قد أكمل تحركه، لذلك قم بالنقر على زر Refresh الموجود في متصفحك لإعادة تحميل الصفحة وتحريك النص مرة أخرى

- alternate يتأرجح النص جيئة وذهاباً من جانب إلى آخر.

```
<MARQUEE BGCOLOR="#FF0000" WIDTH="80%" BEHAVIOR="alternate">
H.P in arabic
</MARQUEE>
```

DIRECTION تحدد إتجاه سير النص وذلك من خلال القيم left التي تحركه إلى اليسار (وهي الإفتراضية) و right التي تحركه إلى اليمين.

```
<MARQUEE BGCOLOR="#FF0000" WIDTH="80%" DIRECTION="right">
H.P in arabic
</MARQUEE>
```

LOOP تحدد عدد المرات التي سيتحرك فيها النص داخل اللافتة. فإذا أردت أن تستمر الحركة إلى ما لا نهاية فضع القيمة 1- أو infinite وبخلاف ذلك ضع عدد المرات التي تريدها. والشيفرة التالية تحدد عدد مرات الحركة بثلاث. (إذا وجدت النص ساكناً أو لم تجد نصاً على الإطلاق فهذا يعني أنه قد استنفذ حركاته الثلاث لذلك تحتاج للنقر على Refresh لإعادة تشغيله)

```
<MARQUEE BGCOLOR="#FF0000" WIDTH="80%" DIRECTION="right" LOOP="3">
H.P in arabic
</MARQUEE>
```

بقي لدينا الآن خاصيتين تقومان بتحديد سرعة تحرك النص في اللافتة وقبل أن نناقشهما دعني أروي لك هذه القصة:

في غابر الأزمان، حدثنا أستاذ الفيزياء يا سادة يا كرام، أن الأفلام المتحركة ما هي إلا مجموعة من الصور الثابتة أو اللقطات التي يتم عرضها بصورة متعاقبة خلال فترة زمنية معينة، مما يعطي الإنطباع بوجود الحركة. وسرعة عرض اللقطات وتعاقبها هو ما يحدد سرعة الحركة للفيلم. فمثلاً مشهد السيارة المسرعة هو مجرد مجموعة من الصور الثابتة لهذه السيارة في مواقع مختلفة على الشاشة. وبما أن لافتاتنا هي شكل من أشكال الأفلام المتحركة فإن هذا المبدأ يسري عليها أيضاً... إلى هنا وانتهت القصة!

والآن إليك الخصائص:

SCROLLAMOUNT: هذه الخاصية تحدد المسافة بين كل لقطة وأخرى للنص وهي تأخذ قيماً عددية صحيحة تحدد المسافة المقطوعة بالبيكسل (أعرف أنها صعبة قليلاً لكن لا بأس سأوضحها بالأمثلة).

```
<MARQUEE SCROLLAMOUNT="1"> HTML </MARQUEE>

<MARQUEE SCROLLAMOUNT="50"> HTML </MARQUEE>

<MARQUEE SCROLLAMOUNT="100"> HTML </MARQUEE>

<MARQUEE SCROLLAMOUNT="200"> HTML </MARQUEE>
```

في الحالة الأولى فإن المسافة التي تقطعها كلمة HTML في كل حركة لها هي 1 بيكسل. أما في المثال الثاني فالمسافة هي 50 بيكسل أي أن الكلمة تقفز 50 بيكسل في كل خطوة (أو لقطة) بما يساوي 50 ضعفاً عن المثال السابق وهذا ما يعطي الإنطباع بالسرعة، وهكذا الأمر للمثالين اللاحقين وهما 100 و 200 بيكسل على التوالي.

SCROLLDELAY: لكي نحدد الزمن الذي يستغرقه النص في القفز بين كل خطوة وأخرى نستخدم هذه الخاصية، والقيمة المعطاه مع هذه الخاصية تمثل الزمن بالميلي ثانية (0.001 من الثانية)
سوف أستخدم نفس الأمثلة السابقة مع هذه الخاصية لنقارن بين النتائج في الحالتين:

```
<MARQUEE SCROLLAMOUNT="1" SCROLLDELAY="500"> HTML </MARQUEE>

<MARQUEE SCROLLAMOUNT="50" SCROLLDELAY="500"> HTML </MARQUEE>

<MARQUEE SCROLLAMOUNT="100" SCROLLDELAY="500"> HTML </MARQUEE>

<MARQUEE SCROLLAMOUNT="200" SCROLLDELAY="500"> HTML </MARQUEE>
```

لقد أسندت القيمة 500 لهذه الخاصية في جميع الأمثلة وهي تعني أن هناك فترة نصف ثانية (بالتمام والكمال) تفصل بين كل خطوة وأخرى للكلمة. وبالمناسبة، قد تضطر للإنتظار دهراً كاملاً لكي ترى النص في المثال الأول.

ما رأيك الآن لو نلغي الخاصية SCROLLAMOUNT لنرى كيف تعمل SCROLLDELAY لوحدها

```
<MARQUEE SCROLLDELAY="500"> HTML </MARQUEE>
```

الوسوم الخاصة 2

Internet Explorer MS

<BODY/> ... <BODY>

هل تعتقد أن هذه وسوماً جديدة أو خاصة تختلف عن تلك التي نعرفها؟! من الواضح أنها ليست بالوسوم الجديدة، بل هي نفسها التي تحدثنا عنها سابقاً ومنذ الدرس الأول. وكل ما هنالك هو أننا سنتعرف على خصائص جديدة لها تختص فقط بــ MS Explorer. ولو بدأت فعلاً في تطبيق الخاصية الأولى لها وهي تدعى BGPROPERTIES... فإنك لو قمت بالنقر على شريط التمرير الجانبي للمتصفح، وأثناء ذلك دققت النظر في خلفية الصفحة ستلاحظ شيئاً مختلفاً؟؟ إن الخلفية ثابتة بينما النص فقط ومحتويات الصفحة هي التي تتحرك فقط.

توضع هذه الخاصية مع الوسم <BODY> وتقوم بتثبيت خلفية الصفحة كما رأينا، وهي تأخذ دائما القيمة FIXED أي أن الشيفرة عموماً تكون

```
<BODY BGPROPERTIES="fixed"> ... </BODY>
```

في أغلب الأحوال، وعندما نريد التحكم بهوامش الصفحة نقوم باستخدام طرق وحيل معينة لأداء هذه المهمة. مثلاً قد نستخدم
 و <P> لإضافة أسطر كما لو كانت هوامش علوية وسفلية للصفحة. كما قد نستخدم <BLOCKQUOTE> ... <BLOCKQUOTE/> وهي وسوم الفقرات المقتبسة لإضافة هوامش جانبية للصفحة أو نستخدم الجداول ... إلخ. لكن يوفر MS Explorer هذا اللف والدوران من خلال دعمه للخصائص التالية في وسم <BODY>

LEFTMARGIN: التي تحدد الهامش الأيسر لكل الصفحة
TOPMARGIN: التي تحدد الهامش العلوي

```
<BODY LEFTMARGIN="50" TOPMARGIN="50"> ... </BODY>
```

وسأترك لك مهمة تجربتها في إحدى صفحاتك

<IFRAME> ... </IFRAME>

من الإمكانيات الرائعة المتوفرة في HTML ويدعمها MS Explorer الإطارات العائمة Floating Frames. كما تعلم لقد تعرفنا على الإطارات بشكل عام في الدروس 9، 10، 11 وتعرفنا على الإمكانيات التي يوفرها هذه النوع من طرق عرض صفحات الويب. وقلنا إن إدراج صفحات الإطارات يتطلب وجود ملف رئيسي جنباً إلى جنب مع الملفات العادية ... إلخ. أما في الإطارات العائمة فالحكاية أبسط بكثير فهي مجرد عناصر تدرج داخل الصفحات مثلها مثل أي عناصر أخرى كالصور، ومن ثم نقوم بتحديد الملف الذي نريد عرضه داخل الإطار بالإضافة إلى بعض الخصائص التي قد نحتاجها. هذا كل ما هنالك!
ولنبدأ بالوسوم الأساسية للإطارات العائمة

```
<IFRAME> Here, there is a floating frame </IFRAME>
```

وهي بحالتها هذه عديمة الفاعلية ولن ينتج عنها أي شيء سوى حجز مساحة فارغة ضمن الصفحة ستخصص لهذا الإطار. و يوجد هناك عبارة مكتوبة داخل الوسوم فماذا بشأنها؟ الحقيقة أن هذه العبارة وضعت بحيث تظهر كبديل عن الإطار في حالة تم عرضه من خلال متصفح لا يدعم الإطارات العائمة. إذن تستطيع رؤية هذه العبارة إذا كنت تستخدم Netscape في مقابل عدم رؤيتك للإطار (حظاً أوفر).

ننتقل مباشرة إلى الخصائص فهي بيت القصيد.

SRC: تحدد الملف الذي نريد إدراجه في الإطار

```
<IFRAME SRC="thissite.html">
Here, there is a floating frame
</IFRAME>
```

WIDTH, HEIGHT: تحددان مساحة الإطار عرضاً وارتفاعاً وكالعادة يتم ذلك باستخدام القيم المطلقة أو النسبية

```
<IFRAME SRC="thissite.html" WIDTH="70%" HEIGHT="200">
Here, there is a floating frame
</IFRAME>
```

SCROLLING: تحدد إذا أردنا إضافة أشرطة التمرير الموجوده على جانب أو أسفل الإطار أم لا، وتأخذ القيم yes (الإفتراضية) أو no

```
<IFRAME SRC="thissite.html" WIDTH="70%" HEIGHT="200" SCROLLING="no">
Here, there is a floating frame
</IFRAME>
```

FRAMEBORDER: تحدد ظهور أو عدم ظهور الحدود ثلاثية الأبعاد حول الإطار وتأخذ القيم 1 (الإفتراضية) و 0 التي تعني لا حدود.

```
<IFRAME SRC="thissite.html" WIDTH="70%" HEIGHT="200" SCROLLING="no" FRAMEBORDER="0">
Here, there is a floating frame
</IFRAME>
```

NAME: من خلال هذه الخاصية نقوم بإسناد اسم للإطار لكي نقوم باستخدامه من قبل الوصلات التشعبية التي نريد استهداف الإطار بها. (راجع <u>الدرس الحادي عشر</u> حول هذا الموضوع)

```
<IFRAME SRC="thissite.html" WIDTH="70%" HEIGHT="200" NAME="myframe">
Here, there is a floating frame
</IFRAME>
```

لقد قمت بتسميته myframe. والآن إذا أدرجت وصلة تشعبية للصفحة الرئيسية –مقدمة في إتش.تي.أم.أل– بالشكل التالي:

```
<A HREF="htmintro.html" TARGET="myframe"> مقدمة في إتش.تي.أم.أل </A>
```

فإن النقر عليها يؤدي إلى تحميل الصفحة داخل هذا الإطار

الدرس السابع عشر
الويب واللغة العربية

سندخل من خلال هذا الدرس إلى أحد المواضيع الهامة التي يُعنى بها كل من يريد أن يكون للغة العربية وجود على الإنترنت. سنقوم بمناقشة الكيفية التي يمكن لنا كمستخدمين عرب إنشاء صفحات ويب بلغتنا على هذه الشبكة العالمية.

فذلكة تاريخية

نحن نعرف أن بدايات اللغة العربية على الإنترنت اقتصرت على أسلوب قد نعتبره الآن بدائياً، لكن لا نستطيع إنكار فعاليته في ذلك الوقت. (لا تعتمد كثيراً على عبارة *"في ذلك الوقت"* فمن المحتمل أن يكون هناك مواقع لا زالت تعتمد هذا الأسلوب لعرض محتوياتها حتى هذه اللحظة)، وهو أسلوب النص المصور. أي النص المسحوب على جهاز Scanner والمعروض على الإنترنت كصورة. وكانت هذه هي الطريقة التي نَفذ منها أصحاب المواقع المعربة للدخول إلى عالم الإنترنت. لكن من الواضح أنها لم تكن بالطريقة الفعالة أو العملية، لعدة أسباب أهمها:
بالنسبة للمصمم: كانت عملية إنشاء الصفحة تعني كتابتها وتنسيقها وطباعتها ومن ثم مسحها على Scanner فإذا أراد فيما بعد إجراء أي عملية تعديل مهما كانت بسيطة فذلك يعني إعادة تحريرها وسحبها، مكلفة في ذلك الوقت والجهد وربما المال.
أما بالنسبة للزائر فذلك يعني إضاعة وقت أطول في عرض هذه النصوص –الصور– أضعافاً مضاعفة مما لو كانت نصوصاً بحتة. بالإضافة إلى استحالة إجراء عمليات البحث المعتادة سواءً كانت خارجية من خلال محركات بحث الإنترنت، أو داخلية ضمن الصفحة نفسها من خلال أمر البحث الموجود في المتصفح. وأحياناً قد يواجه مشكلة رداءة الصورة وبالتالي عدم إمكانية إخراجها بشكل واضح وسليم على الطابعة.

لكن دعنا لا نلم هؤلاء الرواد الذين اتبعوا هذه الطريقة. فكما يقال (مكره أخاك لا بطل) وعذرهم هو عدم وجود متصفحات تستطيع عرض النصوص العربية بطريقة صحيحة. إلى أن قامت شركة صخر بإطلاق متصفحها سندباد الذي جاء معرباً لـــ Netscape وشركة مايكروسوفت التي أطلقت MS Internet Explorer ليكونا فاتحة خير بالنسبة لنا كمستخدمين عرب للإنترنت، ولا نستطيع إلا أن نكنّ لهما كل التقدير. فهانحن كما ترى نبحر في الإنترنت ولغتنا العربية تزداد انتشاراً فيها يوماً بعد يوم.

حسناً، لا تظن أننا سنحتاج إلى إعدادات معينة أو أننا سنتعامل مع وسوم إضافية غير عادية عندما نتحدث عن صفحات ويب باللغة العربية... إطلاقاً، فالواقع أنك تستطيع اعتبار هذا الدرس مجرد دردشة عادية عن الويب واللغة العربية.
ولن يكون هناك إلا وسم جديد واحد سيتم التعرض له بشكل مقتضب بما يهمنا في موضوع اللغة العربية هنا على أن نناقشه بالتفصيل في درس لاحق.

ولنبدأ الآن...

لنقم بداية بتعريف صفحة ويب عادية كما إعتدنا:

```
<HTML>
<HEAD>
<TITLE> ... </TITLE>
</HEAD>

<BODY>
....
</BODY>
</HTML>
```

ومن ثم سنقوم بإضافة هذه الشيفرة لها

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1256">
```

وذلك في القسم الأعلى أي ضمن الوسمين <HEAD> ... </HEAD> ويفضل أن تكتبها بعد وسوم العنوان <TITLE> ... </TITLE>

```
<HTML>
<HEAD>
<TITLE> ... </TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1256 ">
</HEAD>

<BODY>
....
</BODY>
</HTML>
```

قم بكتابة هذه الشيفرة كما هي نصاً وحرفاً. وما يعنينا فيها الآن هي العبارة

charset=windows-1256

فهي لب موضوعنا. أما باقي الشيفرة فسوف نقوم بمناقشتها إن شاء الله في درس لاحق عند الحديث عن الوسوم المتقدمة.

فما الذي تعنيه العبارة السابقة؟؟ بإختصار شديد، هي تحدد قائمة الترميز العربية Code Page التي سيتم عرض صفحتنا من خلالها. أو لنقل أنها تحدد أسلوب التشفير الذي سيستخدمه المتصفح لعرض الأحرف.

حسناً، يبدو أننا سنعود ثانية للفذلكة التاريخية. وهذه المرة إلى بدايات اللغة العربية مع الحاسوب الشخصي نفسه وليس مع الإنترنت فقط:-
إذا كنت من مستخدمي الحاسوب القدامى ... وبالتحديد من أيام نظام التشغيل DOS وتطبيقاته (أي قبل أن يكون هناك برنامج ويندوز المعرّب) فلا بد أنك تعرف، وربما تعاملت مع برامج التعريب القديمة التي واكبت تلك الفترة كبرنامج النافذة وريم ونظم صخر ... إلخ. حيث أنه في تلك الفترة كانت هناك عدة جهات أخذت على عاتقها مهمة تعريب الحاسوب وأنظمة تشغيله. لكن المشكلة كانت في في غياب التنسيق بين هذه الجهات. وكانت النتيجة هي إصدار العشرات من برامج التعريب التي تختلف عن بعضها البعض في **طريقة ترتيب قائمة الرموز والحروف** وبالتالي اختلف أسلوب تشفير النصوص وعرضها. بمعنى أن النص المكتوب من خلال برنامج "النافذة" سوف يبدو كمجموعة من الحروف والكلمات المبهمة وكأنها مكتوبة بلغة أخرى باستخدام برنامج آخر مثل "ريم".

ولتوضيح الفكرة السابقة (على الأقل كما أفهمها أنا!!!) إليك هذا المثال.
لنفرض أن هناك قائمة ترميز رتّبت فيها الحروف بالشكل التالي:

ا	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	هـ	و	ي

وأننا أردنا كتابة كلمة "سندباد" من خلالها...

ثمّ لنفرض الآن أننا قمنا باستعراض الكلمة باستخدام برنامج يتعامل مع قائمة ترميز مختلفة وبالترتيب التالي:

ر	و	ز	ظ	ش	س	ي	ب	ل	ا	ت	ن	م	ك	ط	ذ	ض	ص	ث	ق	ف	غ	ع	هـ	خ	ح	ج	د

بالمقارنة بين النظامين سوف نشاهد أن كلمة "سندباد" في النطام الأول قد أصبحت "ضظفجدف"!!! في النظام الثاني

ي	و	هـ	ن	م	ل	ك	ق	ف	غ	ع	ظ	ط	ض	ص	ش	س	ز	ر	ذ	د	خ	ح	ج	ث	ت	ب	ا
ر	و	ز	ظ	ش	س	ي	ب	ل	ا	ت	ن	م	ك	ط	ذ	ض	ص	ث	ق	ف	غ	ع	هـ	خ	ح	ج	د

إذن تعدد نظم التشفير لم يكن لصالح المستخدم العربي أبداً، إذ أنها أوقعته في إشكالات أقلها عدم التوافق بين التطبيقات التي يتم العمل عليها. ولا شك أن ظهور نظام ويندوز وانتشاره بين المستخدمين العرب بصورة كبيرة قد حدّ من هذه المشكلة بشكل فعّال.

والآن عودة إلى الإنترنت... لقد ألقت المشكلة سالفة الذكر بظلالها على نظم إنترنت ومتصفحاتها المعرّبة (وإن كان ذلك بصورة أقل تأثيراً عمّا سبق). لذلك تجد أن من الميزات التي تحرص شركات صخر وميكروسوفت على إضافتها للمتصفحات هي ميزة تعدد قوائم الترميز التي يستطيع المتصفح دعمها. ومن الأمثلة على هذه القوائم: Windows-1256, DOS-720, ISO-8859-6 وأكثرها استخداماً وانتشاراً هي Windows-1256. وعادة تملك المتصفحات القدرة على اكتشاف قائمة الترميز المطبقة على الصفحة بصورة تلقائية. لكن من الحكمة أن نقوم نحن دائماً بتحديد هذه القائمة ليس بسبب قلة الثقة في المتصفحات وإنما من باب الإحتياط الواجب دائماً في عالم الإنترنت وتصميم الصفحات...

إذن، هل عرفت الآن السبب في إضافة الشيفرة charset=windows-1256 إلى بداية الصفحة؟

لكن ماذا سيحدث لو دخلت إلى إحدى الصفحات ولم تكن تتضمن الشيفرة الخاصة بقائمة الترميز؟ حسناً، هناك إحتمال من إثنين: إما أن يكون المتصفح قد اكتشفها بصورة تلقائية واستخدم الإعدادات الإفتراضية وبالتالي تم عرض الصفحة بالشكل الصحيح وبذلك حلّت المشكلة قبل أن تبدأ. أو تستطيع أنت أن تحل المشكلة بنفسك ، كيف؟؟ إليك الطريقة:

في المتصفح MS Explorer تجد أيقونة صغيرة في الزاوية اليمنى السفلى لنافذة المتصفح. فإذا قمت بالنقر عليها ستظهر قائمة بأسماء قوائم الترميز التي يدعمها.



وما عليك الآن إلا أن تختار القائمة التي تعتقد أنها المناسبة وإذا أخطأت قم باختيار قائمة أخرى وهكذا.

أما في برنامج سندباد فتستطيع أداء هذه المهمة باختيار الأمر "خيارات صفحة الويب الحالية" من قائمة "خيارات سندباد" حيث يظهر بضمنها صندوق حوار خاص بقوائم الترميز.

وبالمناسبة، هل أخبرتك أن باستطاعتك رؤية النص العربي حتى لو كنت تستخدم متصفحاً غير معرب؟ كيف ذلك؟! كل ما عليك فعله هو تغيير إعدادات الخطوط الإفتراضية في المتصفح واختيار خطوط عربية بدلاً من الخطوط المحددة أصلاً. هذه هي الحكاية فقط لكنها ليست الحل!!. إذ سرعان ما ستكتشف أن النص غير مرتب والأسطر والفقرات متداخلة بطريقة غير منطقية خاصة إذا احتوت الصفحة على كلمات أجنبية. فعلى سبيل المثال إذا كانت هناك كلمة

أجنبية في منتصف أحد الأسطر فسوف تلاحظ أن الكلمات التي تليها على اليسار قد أصبحت على اليمين والعكس صحيح. وبذلك سيكون من الصعب عليك متابعة هذه الفقرات. (وعلى أية حال لو كان هذا هو الحل لما وجدنا شركات مثل صخر وميكروسوفت تضيعان وقتهما في إيجاد متصفحات معربة). إذن الحل السابق هو مجرد حل مؤقت لمشكلة ظهور النص العربي فقط ولكنه لم يحل أبداً مشكلة كون اللغتين العربية والإنجليزية تكتبان بإتجاهين مختلفين وبشكل يحتاج إلى حل جذري بإيجاد المتصفحات المعربة ثنائية الإتجاه Bi-Directional أو (BiDi).

لا يخلو الموضوع هنا من الحديث عن الإختلافات بين المتصفحات في التعامل مع صفحات اللغة العربية. (عدنا ثانية للحديث عن الوسوم والخصائص الخاصة بكل متصفح). ولنبدأ بمناقشة تلك الخاصة بالمتصفح ميكروسوفت إكسبلورر:-

في الدرس الرابع تطرقنا إلى وسم الفقرات <P>. وقلنا أن إحدى خصائص هذا الوسم هي الخاصية DIR التي تحدد إتجاه النص من خلال القيم rtl و ltr. (وقد طلبت منك أن تبقى متذكراً لها لأننا سنتطرق إليها عند الحديث عن اللغة العربية.) إذن لقد حان الوقت!!. يدعم إكسبلورر هذه الخاصية بشكل كبير، ذلك لأن من ميزاته أنه لا يتعامل مع الصفحة والفقرات المكونة لها كوحدة واحدة بل أنه يتعامل مع كل فقرة على حدة. ويتطلب هذا أن تقوم بتكرار الإعدادات التي تريدها مع كل فقرة بحد ذاتها.

وعلى سبيل المثال، سوف تجد أن كل فقرة عربية في كل صفحة في هذا الموقع محاطة بالوسوم

```
<P DIR="rtl" ALIGN="right"> ... </P>
```

وأن كل فقرة إنجليزية محاطة بالوسوم

```
<P DIR="ltr" ALIGN="left"> ... </P>
```

وذلك ليس لسبب إلا لضمان ظهورها بالشكل المناسب في المتصفح إكسبلورر. ولاحظ أنني في الحالتين أقوم بتحديد إتجاه النص جنباً إلى جنب مع المحاذاة المطلوبة. وفي الحقيقة أن هذه الخاصية DIR تستخدم ليس فقط مع وسوم الفقرات بل مع أي وسوم أخرى نستخدم معها الخاصية ALIGN وبشكل خاص مع الجداول.

نأتي الآن إلى المتصفح سندباد... كما تعلم فإن هذا المتصفح هو برنامج مضاف Plugin لتعريب المتصفح Netscape بإصداراته المختلفة. لذلك من البديهي أن يرث خصائصه المختلفة، إبتداءاً من الواجهة التطبيقية للبرنامج وإنتهاءاً بالوسوم الخاصة التي يدعمها. لذلك فعند الحديث عن أحدهما نستطيع القول بأننا نتحدث عن الآخر بدون أدنى اختلاف.

ما يهمنا في هذا المقام هو أن سندباد يتعامل مع الصفحة التي يعرضها كوحدة واحدة، وذلك على خلاف إكسبلورر. بمعنى أنه لا يدعم الخاصية DIR. لذلك نرى أن سندباد يحتوي على زر خاص بتحويل إتجاه الصفحة يميناً أو يساراً، وهو الزر الموجود في أعلى نافذته ويكون على شكل سهم.

إذن بشكل عام، طالما كانت الصفحة موحدة اللغة (سواءاً كانت هذه اللغة عربية أو إنجليزية) وبالتالي كانت تحتوي على فقرات موحدة الإتجاه (يميناً أو يساراً) فمن غير الضروري أن نقوم بتحديد المحاذاة لكل فقرة على حدة، فالأصل أن نقوم بذلك من خلال زر تحديد الإتجاه. لكن عندما تحتوي صفحتنا العربية على فقرة إنجليزية (أو العكس) فهنا لا بد من استخدام وسم الفقرات وتحديد المحاذاة المطلوبة لهذه الفقرة فقط، ومن ثم المتابعة مع باقي الفقرات العربية بشكل طبيعي كالسابق. ونكرر العملية مع أي فقرة إنجليزية لاحقة. وهنا عند الضغط على زر تحديد الإتجاه ستجد أنه يغير إتجاه جميع الفقرات فالعربية ستحول إلى اليسار والإنجليزية إلى اليمين والعكس بالعكس.

السؤال الآن، أي من الطرق نستخدمها في صفحاتنا؟ هل نعتمد على الخاصية التي يدعمها إكسبلورر أو على الزر الذي يتضمنه سندباد؟

الجواب عموماً هو أن نتبع الطريقة التي تضمن أكبر قدر من التوافق في مظهر الصفحات فوجود الخاصية DIR لن يضر بصحة المتصفح سندباد. لكن غيابها هو ما سيجعل إكسبلورر هزيلاً. إذن لنستخدمها دائماً ومع كل فقرة من فقرات صفحتنا.

خلاصة القول أضعها في النقاط التالية، وكما يقال: في الإعادة إفادة!

1. **في بداية الصفحة ضع الشيفرة التالية كما هي نصاً وحرفاً**

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1256">
```

2. **قم بإحاطة كل فقرة عربية من فقرات صفحتك بالوسوم التالية:**

```
<P DIR="rtl" ALIGN="right"> ... </P>
```

3. **إذا أردت إدراج فقرات إنجليزية فقم بإحاطة كل منها بالوسوم**

```
<P DIR="ltr" ALIGN="left"> ... </P>
```

4. **تستطيع استخدام الوسوم السابقة مع الجداول أيضاً**

في الأيام الأولى لإنشاء هذا الموقع، وبالذات في الدرس الأول، واجهت إشكالاً معيناً وذلك عندما كنت أقوم بكتابة بعض المصطلحات الإنجليزية وبضمنها رموز خاصة أو أرقام. فقد كانت هذه الرموز تظهر بصورة معكوسة عندما يتم عرضها في المتصفح سندباد. مثال ذلك: أن الإشارة / الموجودة في وسوم النهاية كانت تظهر في آخر الوسم (أي إلى اليمين) بدلا من أن تظهر في بدايته. كذلك عند الحديث عن الرموز الخاصة كالفراغات كانت إشارة (&) تظهر على اليسار وإشارة (;) تظهر على اليمين، أي انهما تظهران بشكل معاكس لما يجب أن يكونا عليه.

وبصراحة مطلقة فقد سببت لي هذه المشكلة كابوساً على مدى عدة أيام متلاحقة، حاولت خلالها تجربة العديد من الحلول. كان من ضمنها أن أقوم بكتابة هذه الرموز بصورة معاكسة علّها تعود إلى رشدها. لكن للسخرية كانت في الحالة المعكوسة تبقى كما هي في مكانها. وأخيراً جاء الحل السحري بعد أن أرسلت إلى شركة صخر أستفسر عن هذه المشكلة. وكان هذا الحل بكتابة رمز يسمى OXFD وهو رمز الفراغ باللاتيني، وتتم كتابته بالضغط على المفتاح ALT في لوحة المفاتيح مع الأرقام (من اليسار إلى اليمين) 0253 وذلك قبل أي رمز "مشاغب" لا يظهر في مكانه الصحيح. وبذلك تم حل كل الإشكالات التي واجهتها مع هذه الرموز.

إذن أستطيع الآن أن أضيف نقطة خامسة إلى النقاط السابقة:

استخدم ALT+0253 قبل الرموز أو الحروف الإنجليزية التي لا تظهر في مكانها الصحيح عند عرضها من

الدرس الثامن عشر
وسوم META ومحركات البحث

من المحتمل أن يكون عنوان هذا الدرس غريباً بعض الشيء بالنسبة لك. وأنه قد أثار لديك بعض التساؤلات عن فحوى هذه الوسوم وعن الهدف من هذا الدرس بشكل عام... ولكي لا تطول تساؤلاتك دعني أوضحها لك كما يلي:-

لقد تابعنا معا من خلال الدروس السابقة مفردات لغة HTML خطوة بخطوة. ومن المحتمل أنك قمت بتطبيق هذه الدروس بصورة عملية من خلال صفحات وهمية قمت أنت بإنشائها. (حسناً فعلت!!! – لأنه من الصعب استيعاب أي موضوع جديد دون القيام بالتطبيق العملي له، وخصوصاً في مجال الكمبيوتر عموماً ولغاته بشكل خاص). والآن أنت تعتقد أنه حان الوقت للعمل الجدي، فلديك فكرة لموضوع ما، وتريد طرحها على الإنترنت. وأنك تفكر بإنشاء موقعك الخاص لتعرض هذا الموضوع من خلاله. وتؤمن أن هذه الدروس التي تتابعها في هذا الموقع قد كوّنت لديك المعرفة الكافية للقيام بتصميم موقعك بنفسك (كما أتمنى)... وبذلك تجد نفسك جاهزاً للانطلاق في رحاب الإنترنت ولتقول بكل فخر: "لدي موقع على الإنترنت". لكنك أيضا قد تتوقف لتفكر للحظات... فأنت تعرف والجميع يعرف أنه يوجد في الإنترنت حالياً الملايين من المواقع! وربما تتساءل: كيف لي أن أنشئ موقعي في هذا الخضم الهائل من المواقع؟! وكيف للناس أن يجدوا طريقهم إلى هذه الزاوية الصغيرة التي أنشأتها في هذا الكيان اللامحدود؟ وإذا كنت من النوع المتشائم فربما ستتراجع عن هذه الفكرة (فكرة الموقع) قبل أن تبدأها... حسناً، لا أريدك أن تشعر بالإحباط. فالطبع لو أن كل واحد منّا يفكر بالجدوى من إنشاء موقع إنترنت خاص به بهذه الطريقة لما وجدت كل هذه المواقع ولما تكونت الإنترنت أصلاً، أليس كذلك؟

في بدايات الإنترنت، كانت هناك عبارة دارجة تقول

Build it, and they will come

أي أنشيء موقعك، وسوف يأتون إليه. ولا أعرف إلى أي مدى كان صدق هذه العبارة في تلك الأيام. فأنا لا أستطيع تخيل أي موقع على الويب تهلّ إليه جحافل الزوار بمجرد افتتاحه مهما كانت طبيعة هذا الموقع أو موضوعه أو التقنيات المستخدمه فيه. لكن ما أعرفه حالياً أن تسويق الموقع في متاهات الويب يعد من أهم وأصعب المهمات التي تواجه أصحابه. بل وربما أصعب من عملية إنشاء الموقع نفسها.

السؤال الآن ... ما هي الطريقة المثلى للوصول إلى شيء ما عندما تحتاجه في الإنترنت؟ والجواب بديهي، إنها محركات البحث. لذلك لا نبالغ إذا قلنا أن الموقع الذي يدرج في محركات البحث وبالذات في الصفحات الأولى منها وفي أعلى مستويات هذه الصفحات هو موقع ناجح يستطيع أن يضمن وصول أكبر عدد من الزوار إليه.

إذن فمحركات البحث هي الخطوة الأولى

لكن كيف نصل أصلاً إلى محركات البحث هذه؟ (وأعني هنا كيف نتوصل إلى إدراج موقعنا في قوائم هذه المحركات؟) وهو ما سنحاول الإجابة عليه في هذا الدرس.

حسناً... قد تقول: نحن هنا نناقش لغة HTML فما دخل محركات البحث بهذا الموضوع؟ بالطبع فإن الهدف من هذا الدرس هو إكمال ما بدأناه في الدروس السابقة حول لغة HTML أي أننا سوف نناقش وسوماً جديدة لهذه اللغة. لكن بما أن هذه الوسوم تتعلق بمحركات البحث. فقد وجدت أنه من المحتم توضيح العديد من النقاط حول هذه المحركات. وعلى أية حال فالمزيد من المعرفة لن يضر أحداً.

إذن فما ستجده هنا في هذا الدرس لن يكون فقط وسوم HTML بل أيضاً بعض المعلومات والأفكار حول محركات البحث، ولن يعنينا هنا كيفية استخدامها للبحث بل ما سيعنينا هو كيفية إدراج المواقع ضمنها... أي أنك لن تكون هنا باحثاً عن المعلومة، بل ستكون ناشراً لها. وأتمنى أن تكون هذه الأفكار عوناً لك على ذلك.

كيف لك أن تجعل الناس يأتون إلى موقعك، وأن يجدوا طريقهم إليه بسهولة ويسر؟ وبشكل أعمّ ... كيف تستطيع أن تسوّق موقعك؟؟؟

هذا هو السؤال

هنا لن أتكلم معك في أمور بديهية وأقول لك إن الفكرة الموجودة في موقعك والموضوع المطروح فيه لهما أثر واضح في جذب الزوار إليه. ولن أقول لك بأن طريقة تنظيمه وتنسيقه تسهل عليهم التجول فيه، وبالتالي لن يجدوا صعوبة في إيجاد ما يبحثون عنه. كذلك لن أذكّرك بأن التجديد المستمر لمحتوياته يجعل زوارك يحرصون على العودة مرة بعد مرة... طبعا لن أقول لك كل ذلك!!!!

لا أعرف مدى استخدامك لمحركات البحث للوصول إلى ما تريده في الإنترنت! ولا أعرف أيضاً إن كان استخدامها قد أثار لديك بعض التساؤلات! على سبيل المثال:

- ما الفرق بين Yahoo و Altavista؟
- ما السبب في تباين النتائج التي تظهر من موقع لآخر من حيث العدد والمحتوى؟
- لماذا يوجد وصلات تشعبية لا تعمل في بعض المحركات أكثر من غيرها؟

عند الحديث عن مواقع البحث في الإنترنت، من الضروري أن نميز بين محركات البحث من جهة والفهارس (أو أدلة البحث) من جهة أخرى، وذلك من حيث المفهوم والخصائص وأسلوب العمل.

محركات البحث Search Engines

يعتبر محرك Altavista من أوضح الأمثلة على محركات البحث. وهي تسمى أيضاً بالزواحف Crawlers أو العناكب Spiders ومردّ هذه التسميات إلى الطريقة التي يتبعها المحرك لإيجاد المواقع وتصنيفها حيث تقوم الروبوتات الخاصة به (برامج خاصة بالبحث وتتميز بالذكاء ...الإصطناعي طبعاً) بالزحف في الإنترنت بحثاً عن المواقع الجديدة وعندما تجد أحدها تقوم بفهرسة وتصنيف كل كلمة من الكلمات الموجودة في صفحات هذا الموقع ضمن قاعدة البيانات الهائلة الخاصة بالمحرك، حيث تستخدم هذه البيانات لتكوّن نتائج البحث التي يطرحها زوار المحرك فيما بعد. وتقوم الروبوتات أيضاً من حين لآخر بإعادة زيارة المواقع السابقة بعد مدة قد تطول أو تقصر (أسبوع أو شهر) تبعاً لطبيعة المواقع نفسها وعدد الزوار اليوميين وكبر الحجم وذلك بهدف تحديث المعلومات عن هذه المواقع وإعادة تصنيف أي إضافات أو إلغاءات طرأت عليها. وكل ذلك يجرى بصورة أتوماتيكية من خلال هذه الروبوتات. ونستطيع القول أن التصنيفات الموجودة في مثل هذه المحركات تكون إلى حد ما حديثة.

الفهارس أو أدلة البحث Directories

ولعل من أبرز الأمثلة عليها هو الفهرس Yahoo وفي هذا النوع من طرق البحث يتم تصنيف المواقع ضمن قوائم متدرجة ومتشعبة عن بعضها البعض بحيث يبدأ بالمفتاح الأساسي العام ثم يتدرج إلى الأكثر تحديداً وهكذا. ويقوم بعملية التصنيف هذه طاقم يعمل خصيصاً في البحث عن المواقع. (أي طاقم بشري، وذلك على خلاف المحركات) كما يعتمد على المعلومات التي يقوم أصحاب المواقع بإرسالها إلى الفهرس عندما يقومون بتحميل المعلومات الخاصة بموقعهم إليه. لذلك نجد أن كمية المعلومات المصنفة في الأدلة أقل منها في محركات البحث وأحياناً أقل حداثة.

ولكي أوضح لك الفرق بين كلا الموقعين إليك المثال العملي التالي:

عندما قمت بالبحث عن المواقع الخاصة بمدينتي نابلس باستخدام Yahoo كانت نتيجة البحث ثلاثة مواقع فقط، وكانت إحدى النتائج كما يلي:

Regional: Regions: Middle East: Countries and Regions: Palestinian Authority: Cities and Regions: Nablus

وكما ترى فإن تسلسل البحث انتقل من Regional إلى Region إلى Middle East وهكذا نزولاً إلى أن وصل إلى نابلس

أما باستخدام Altavista فكانت النتيجة حوالي 4218 موقع، حيث تمّ سرد المواقع تباعاً ودون أي تصنيف أو ترتيب محدد. والحقيقة أن هناك صفحات في نفس الموقع تكرر ظهورها ضمن نتائج البحث، والسبب في ذلك فقط لأن كلمة نابلس تكررت في كل صفحة من صفحات الموقع.
وطبعاً كما تلاحظ فإن هناك فرقاً شاسعاً بين النتائج التي تم الحصول عليها ومردّه كما أسلفت إلى طريقة العمل والتصنيف المتبعة في كل من المحركين.

والآن لنبدأ حديثنا عن وسوم META فما هي هذه الوسوم ؟؟؟
باختصار شديد، هي وسوم تدرج ضمن صفحات الويب، وبالتحديد في أعلاها وضمن المقطع <HEAD> ... </HEAD> وتستخدم لوصف الصفحة أو الموقع بشكل عام، من حيث المحتويات والكلمات الرئيسية أو المفتاحية والمؤلف وغيرها من المعلومات التي قد نعتبرها توثيقية. كما أنها تفيد في عملية التصنيف التي تجرى من قبل محركات البحث. (ولاحظ أنّي قلت محركات البحث ولم أقل أدلة البحث) فالحقيقة أن محركات البحث مثل Altavista, Hotbot, Infoseek هي التي تستفيد فقط من هذه الوسوم وتلجأ إليها في عملية تصنيف الموقع ولا تستفيد منها الأدلة. أي أن Yahoo لا يتعامل معها أبداً. ومع ذلك فإن نشرك لصفحاتك على الإنترنت دون وضع هذه الوسوم ضمنها يعتبر إغفالاً لجانب مهم من جوانب إنشاء المواقع على الويب. **ولكن ...** (عادة عندما تظهر "لكن" في منتصف الحديث، فإن هناك أخباراً غير سارة في الطريق) لا تعتبر هذه الوسوم حلاً سحرياً لمسألة محركات البحث، صحيح أنها تزيد من إحتمالات إدراج موقعك في هذه المحركات، وأنها تساعد المحركات على تصنيف صفحاتك بالطريقة التي تراها أنت مناسبة. لكن سيبقى أمامك خطوات أخرى لإكمال هذه المهمة.

لقد حان الوقت للتّكلم بلغة HTML... وأبدأ بالقول إن الوسم <META> هو وسم مفرد. وهو يأخذ الخصائص التالية:

HTTP-EQUIV
تعتبر هذه الخاصية مكافئةً للوسم <HEAD> الذي يعرف الصفحة ككل. (والحقيقة أن اسمها يدل على ذلك) وبدون الدخول في تفاصيل جانبية، نستطيع أن نقول أنها تحدد خصائص الصفحة ككل.

وإذا كانت لديك ذاكرة قوية فسوف تدرك أننا قمنا بالفعل باستخدام هذه الوسوم مسبقاً!! وكان ذلك في الدرس السابق عندما حددنا صفحة الترميز العربية التي سيتم عرض الصفحة من خلالها وكان ذلك بالشكل التالي:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1256">
```

NAME
نقوم من خلال هذه الخاصية بتحديد اسم (أو لنقل عنوان) وحدة البيانات التي نريد تعيين قيمها
CONTENT

تحدد المحتويات أو البيانات التي نريد إسنادها للخاصيتين السابقتين

في واقع الأمر فإن لهذه الخصائص الكثير من القيم. لكنها بكل بساطة لا تعنينا في شيء، فمعظم هذه القيم توصف بأنها Server Side أي يتم التعامل معها على مستوى المزود وليس على مستوى جهاز المستخدم Client Side ولا تهم مصممي صفحات بسطاء مثلنا. وما يعنينا بالدرجة الأولى هي القيم التالية:

- مع الخاصية HTTP-EQUIV...
  القيمة Content-Type والتي تعني نوعية المحتويات. وأذكرك مرة أخرى أننا تعاملنا معها مسبقاً ... ثم باستخدام الخاصية Content قمنا بتحديد طبيعة هذه المحتويات على أنها text/html أي نصوص لغة HTML وأن صفحة ترميزها هي Windows-1256 ...ولن نحتاج أكثر من ذلك مع هذه الخاصية.
- مع الخاصية NAME... نستخدم
  - القيمة keywords لتحديد الكلمات المفتاحية للصفحة.

```
<META NAME="keywords" ... >
```

ثم نستخدم الخاصية Content لتحديد هذه الكلمات. وعلى سبيل المثال إليك الشيفرة التالية التي أستخدمها في صفحات هذا الموقع

```
<META NAME="keywords"CONTENT="html,webpage design,psp,paint shop pro,arabic site">
```

ولا ضرر في تكرار بعض أو كل الكلمات المفتاحية لثلاث أو أربع مرات للتأكيد عليها

```
<META NAME="keywords"CONTENT="html,html,html,html,webpage design,psp,paint shop
pro,arabic site">
```

  - القيمة description لتوصيف الصفحة، وذلك بعبارات قصيرة تلخص محتويات الصفحة والهدف منها.

```
<META NAME="description" CONTENT="Html and Paint Shop Pro tutorials in Arabic,learning web
page design in Arabic">
```

  - القيمة author لتوثيق اسم المؤلف أو صاحب الموقع

```
<META NAME="author" CONTENT="Abu Al-Abed">
```

  - القيمة copyright لسرد حقوق النشر الخاصة بالصفحة

```
<META NAME="copyright" CONTENT="....">
```

هذه هي وسوم META التي تهمنا. وسوف أقوم الآن بإجمالها لك في شيفرة واحدة مع باقي الوسوم في ترويسة الصفحة بحيث يمكنك قصها ولصقها في كل صفحة من صفحاتك. وعليك طبعا كتابة القيم والبيانات التي تريدها فيما بعد حسب عنوان صفحتك والكلمات المفتاحية والوصف الذي تريده (لا تتوقع مني أن أقوم أيضاً بكتابه هذه القيم لك). كذلك لا تنس تحديد صفحة الترميز إن كانت مختلفة أو حذف الوسم الخاص بها إن كانت صفحتك باللغة الإنجليزية.

```
<HEAD>
<TITLE> ... </TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1256">
```

```
<META NAME="keywords" CONTENT=" ... ">
<META NAME="description" CONTENT=" ... ">
<META NAME="author" CONTENT=" ... ">
<META NAME="copyright" CONTENT=" ... ">
</HEAD>
```

هل هذا يكفي؟ وأعني هل تكفي هذه الوسوم لتكون من الأوائل دائماً على صفحات محركات البحث؟ كما أسلفت هناك دائماً جهود إضافية يجب عليك بذلها. وهذه المرة عليك التوجه إلى هذه المحركات والبحث عن الوصلات التشعبية الخاصة بإضافة المواقع الجديدة. في Altavista ستجد أنها تسمى ***Add Page***. وفي Infoseek و Excite و HotBot ستجدها ***Add URL*** أما في Yahoo فهي ***How To Suggest a Site?*** كذلك ستجد مثل هذه الوصلات في المواقع العربية وعادة تعرف بـِ ***أضف موقعك.***

على أية حال قم بالنقر عليها لكي تقودك إلى نموذج بسيط عليك ملؤه ببعض المعلومات عن موقعك (لا تخف فإن طريقة التعامل مع هذه النماذج هي أبسط مما تتصور). لكن لا تتوقع أن ترى موقعك وقد تمت إضافته في الدقائق التالية لهذه العملية، فهذا يحتاج في معظم الأحوال إلى عدة أيام.

أعرف أنك ستقول عن هذه العملية بأنها صعبة وشاقة... لذلك سوف أدلك على طريقة أسهل وأسرع بكثير. هناك برامج خاصة لإدراج المعلومات عن المواقع والصفحات بحيث تقوم بملء نموذج خاص لمرة واحدة فقط بكل ما تريده من معلومات ثم يقوم هذا البرنامج بإرسالها إلى المئات من محركات وأدلة البحث... هكذا ببساطة وبدقائق معدودة. ومن هذه البرامج برنامج AddWeb وبرنامج Submission Wizard لكن من المؤسف أن النسخ المشتركة لهذه البرامج أو مثيلاتها، تكون محدودة الفعالية أي تتعامل مع عدد محدود من المحركات والأدلة وليس كلها إلا إذا قمت بشراء هذه البرامج. ...لكن لا بأس من تجربتها على الإطلاق.

وأخيرا أتمنى أن تكون قد استفدت من هذا الكتاب وأنصحك بزيارة هذا الموقع فهو مفيد جداً في هذا المجال

http://www.khayma.com/hpinarabic/